



Contents lists available at [ScienceDirect](#)

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



Explicitly antisymmetrized neural network layers for variational Monte Carlo simulation

Jeffmin Lin ^{a,*}, Gil Goldshlager ^a, Lin Lin ^{a,b,*}

^a Department of Mathematics, University of California, Berkeley, CA 94720, USA

^b Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA



ARTICLE INFO

Article history:

Received 9 December 2021

Received in revised form 24 September 2022

Accepted 7 November 2022

Available online 12 November 2022

Keywords:

Variational Monte Carlo

Antisymmetry

Electronic structure

Machine learning

Neural networks

ABSTRACT

The combination of neural networks and quantum Monte Carlo methods has arisen as a promising path forward for highly accurate electronic structure calculations. Previous proposals have combined equivariant neural network layers with a final antisymmetric layer in order to satisfy the antisymmetry requirements of the electronic wavefunction. However, to date it is unclear if one can represent antisymmetric functions of physical interest, and it is difficult to precisely measure the expressiveness of the antisymmetric layer. This work attempts to address this problem by introducing explicitly antisymmetrized universal neural network layers. This approach has a computational cost which increases factorially with respect to the system size, but we are nonetheless able to apply it to small systems to better understand how the structure of the antisymmetric layer affects its performance. We first introduce a generic antisymmetric (GA) neural network layer, which we use to replace the entire antisymmetric layer of the highly accurate ansatz known as the FermiNet. We demonstrate that the resulting FermiNet-GA architecture can yield effectively the exact ground state energy for small atoms and molecules. We then consider a factorized antisymmetric (FA) layer which more directly generalizes the FermiNet by replacing the products of determinants with products of antisymmetrized neural networks. We find, interestingly, that the resulting FermiNet-FA architecture does not significantly outperform the FermiNet. This strongly suggests that the sum of products of antisymmetries is a key limiting aspect of the FermiNet architecture. To explore this further, we investigate a slight modification of the FermiNet, called the full determinant mode, which replaces each product of determinants with a single combined determinant. We find that the full single-determinant FermiNet closes a large part of the gap between the standard single-determinant FermiNet and FermiNet-GA on small atomic and molecular problems. Surprisingly, on the nitrogen molecule at a dissociating bond length of 4.0 Bohr, the full single-determinant FermiNet can outperform the largest standard FermiNet calculation with 64 determinants.

© 2022 Elsevier Inc. All rights reserved.

* Corresponding authors.

E-mail addresses: jeffminlin@berkeley.edu, jeffmin.lin@gmail.com (J. Lin), linlin@math.berkeley.edu (L. Lin).

1. Introduction

A fundamental challenge in modeling the behavior of electrons in the many-body Schrödinger equation is that the electronic wavefunction must be antisymmetric with respect to particle exchange. This constraint is due to the Pauli exclusion principle and applies to any fermionic wavefunction. When the number of electrons grows, effective parametrization of the space of such wavefunctions becomes difficult. Deep learning techniques have recently impacted *ab initio* quantum chemistry by providing a new approach to the problem of tractable parameterization of high dimensional function spaces in quantum many-body problems. Over the past few years, a growing number of works [1–11] have demonstrated the use of neural networks in wavefunction approximation, with an increasing amount of importance placed on building symmetry constraints into models. In particular, several works [5,6,8,9,11] have recently applied neural networks to model antisymmetric wavefunctions.

The simplest ansatz for representing antisymmetric electronic wavefunctions is known as a Slater determinant, which is an antisymmetrized product of single particle orbitals. The optimization of this ansatz is the core of the Hartree-Fock (HF) method [12]. Conventionally, the representation power of the Slater determinant has been improved by including multiplicative Jastrow factors and transforming the particle coordinates via a so-called backflow transformation [13,14], resulting in the Slater–Jastrow–backflow ansatz. While the Hartree Fock problem can be solved efficiently for a wide range of systems of interest using matrix diagonalization methods, the Slater–Jastrow and the Slater–Jastrow–backflow ansatzes are significantly more complicated and can in practice only be optimized using the quantum Monte Carlo (QMC) method known as variational Monte Carlo (VMC) [15–18]. According to the variational principle, the energy obtained by any admissible wavefunction ansatz is lower bounded by the exact ground state energy. For strongly correlated quantum systems, and even weakly correlated quantum systems when high accuracy is required, a linear combination of either a large number of Slater determinants (called the configuration interaction method (CI)) or a number of Slater–Jastrow–backflow ansatzes is needed to yield a sufficiently low, and therefore accurate, energy estimate.

Recently, these considerations have led to an active interest in leveraging neural networks to improve the construction of the backflow [5,8,9], the antisymmetry [6,9], and the Jastrow factor [8] of these ansatzes. PauliNet [8] uses relatively small permutation equivariant neural networks for the backflow and invariant neural networks for the Jastrow factor. The backflow transformation in PauliNet is applied multiplicatively to the Hartree-Fock orbitals before the determinant layer is applied. The FermiNet work [9,19], revealed around the same time as PauliNet, uses a more sophisticated equivariant backflow transformation with many more parameters. Another interesting and surprising feature of the FermiNet is that it eschews the Jastrow factor entirely. For a given system, the FermiNet often achieves lower energies than PauliNet [19].

While there has been some progress [20–24] in analyzing the expressiveness of the permutation equivariant mappings used in the backflow construction [25], the understanding of the effectiveness of the antisymmetric neural network layers remains limited [20,23,26]. Interestingly, Refs. [5,9,23] propose that a single FermiNet determinant could in theory achieve a universal representation of antisymmetric functions. However, these constructions are based on either a sorting process [5,9] or an equivariant mapping that essentially encodes the entire wavefunction [23]. Both constructions yield discontinuous feature mappings when the ambient space dimension is larger than 1 or the number of particles is greater than 2, as opposed to the continuous neural network layers used in all works in the literature so far. The success of both PauliNet and FermiNet also depends crucially on the quality of the permutation equivariant backflow. Therefore when the VMC energy is higher than the exact ground state energy, it is difficult to pin down the source of the error.

To address this issue, we consider wavefunction ansatzes which replace parts of the antisymmetric layer of FermiNet with explicitly antisymmetrized universal neural networks. The obvious drawback of this approach is that its computational cost increases factorially with respect to the number of electrons, and it can therefore only be applied to very small atoms and molecules. However, the use of explicit antisymmetrization can still allow us to better understand how the structure of the antisymmetric layer affects the overall performance.

We first consider a generic antisymmetric (GA) layer, which replaces the entire sum of products of determinants structure in FermiNet with an explicitly antisymmetrized feed forward neural network. When combined with the FermiNet backflow, the resulting FermiNet-GA architecture can achieve a universal representation of antisymmetric functions by construction. We also find that the FermiNet-GA structure is empirically a highly expressive ansatz. For all systems studied, the error of the correlation energy is less than 1%, and is well below chemical accuracy ($1 \text{ kcal/mol} \approx 1.6 \times 10^{-3} \text{ a.u.}$). On the other hand, at least from a practical perspective, we find that the so-called single-determinant FermiNet, which is in fact computed as a product of two determinants, is not expressive enough to represent electronic wavefunctions of interest.

To investigate further, we replace the product of determinants in the single-determinant FermiNet with a product of explicitly antisymmetrized feed forward neural networks, yielding the factorized antisymmetric neural network layer of rank 1 (FA-1). We find that FA-1 is not able to outperform single-determinant FermiNet, which suggests that the ineffectiveness of the single-determinant FermiNet is closely related to its product structure. We define a factorized antisymmetric neural network layer of rank K (FA- K), which generalizes the structure of K -determinant FermiNet. We find that FermiNet-FA- K does not outperform K -determinant FermiNet, which indicates that the sum-of-products structure is a key limiting feature of both architectures.

These results suggest that removing this sum-of-products structure may be a promising avenue towards developing an efficient antisymmetric layer that is more expressive than the original FermiNet architecture. We thus study a variant of the FermiNet called the full determinant mode, which replaces the products of determinants used in the FermiNet with single

combined determinants. The full determinant construction was mentioned in the original FermiNet paper [9] and was later implemented by the same authors in the JAX branch of the FermiNet repository [27], but to our knowledge its performance has not been previously reported in the literature. We specifically investigate the full single-determinant FermiNet, which replaces the product of two determinants used in single-determinant FermiNet with a single combined determinant. Our numerical results show that full single-determinant FermiNet can close a large part of the gap between standard single-determinant FermiNet and the true ground state energy on small atomic and molecular problems. We further evaluate the performance of full single-determinant FermiNet on the nitrogen molecule at a dissociating bond length of 4.0 Bohr, a challenging strongly correlated system where the standard FermiNet architecture is not able to yield accurate results even with 64 determinants. We find that the full single-determinant FermiNet can outperform the standard 64-determinant FermiNet on this system, and the error of the energy can be as small as 0.4 kcal/mol compared to r_{12} -MR-ACPF method.

2. Preliminaries

2.1. Many-body electron problem

We consider isolated quantum chemical systems in \mathbb{R}^3 and use atomic units (a.u.) throughout the paper. Let $\{\mathbf{r}_i\}_{i=1}^N$ represent electron positions, $\{\mathbf{R}_I^{(a)}\}_{i=1}^M$ nuclei positions, and Z_I the I th nuclear charge. Under the Born-Oppenheimer approximation, the many body Hamiltonian with M nuclei and N electrons is

$$H = \hat{T} + \hat{V}_{ei} + \hat{V}_{ee} + E_{II}, \quad (1)$$

where the Hamiltonian is partitioned into the kinetic, electron-ion potential, electron-electron, and ion-ion interaction, respectively:

$$\hat{T} = \sum_{i=1}^N -\frac{1}{2} \Delta_{\mathbf{r}_i}, \quad \hat{V}_{ei} = -\sum_{i=1}^N \sum_{l=1}^M \frac{Z_l}{|\mathbf{r}_i - \mathbf{R}_l^{(a)}|}, \quad \hat{V}_{ee} = \sum_{i < j}^N \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad E_{II} = \sum_{I < J}^M \frac{Z_I Z_J}{|\mathbf{R}_I^{(a)} - \mathbf{R}_J^{(a)}|}. \quad (2)$$

For a given atomic configuration $\{\mathbf{R}_I^{(a)}\}$, the ion-ion interaction simply adds a constant shift to the Hamiltonian.

Let $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ be the collection of spatial and spin indices of N electrons, where $\mathbf{x}_i = (\mathbf{r}_i, \sigma_i)$, and assume the N electrons can be partitioned into N_\uparrow spin-up electrons and N_\downarrow spin-down electrons. We denote by $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_N)$ the collection of spatial coordinates of the electrons. We also denote by $\mathbf{R}^{(a)} = (\mathbf{R}_1^{(a)}, \dots, \mathbf{R}_M^{(a)})$ the collection of atomic positions. Then the electron many-body wavefunction $\Psi(\mathbf{X}) \equiv \Psi(\mathbf{x}_1, \dots, \mathbf{x}_N)$ is required to be antisymmetric with respect to the natural action of the symmetric group S_N on the electron configuration \mathbf{X} (i.e., the permutation of the particle indices):

$$\Psi(\pi(\mathbf{X})) \equiv \Psi(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(N)}) = (-1)^\pi \Psi(\mathbf{X}), \quad (3)$$

where $(-1)^\pi$ is the sign or parity of the permutation $\pi \in S_N$.

Note that the Hamiltonian in Eq. (1) does not explicitly depend on the spin. As a result, the antisymmetry constraint over the spatial-spin electron configuration can be rewritten using a spin-independent wavefunction $\Psi(\mathbf{R})$ (abusing notation to reuse Ψ), which reduces the number of degrees of freedom in the wavefunction and improves the efficiency in quantum Monte Carlo calculations [15].

To see this, let \hat{O} be any totally symmetric spin-independent operator (e.g. electron density, energy), and assume we are interested in expectation values of \hat{O} with respect to the many body wavefunction $\Psi(\mathbf{X})$:

$$\langle \hat{O} \rangle = \frac{\sum_\sigma \int \Psi^*(\mathbf{X}) \hat{O}(\mathbf{R}) \Psi(\mathbf{X}) d\mathbf{R}}{\sum_\sigma \int \Psi^*(\mathbf{X}) \Psi(\mathbf{X}) d\mathbf{R}}. \quad (4)$$

For each spin-configuration $\sigma = (\sigma_1, \dots, \sigma_N)$, we can always permute the particle index $1, \dots, N$ so that the spin-up indices appear in front of the spin-down indices. The integrals in the numerator and denominator of Eq. (4) are independent of such a permutation operation. Therefore we may define the spatial wavefunction

$$\Psi(\mathbf{R}) = \Psi((\mathbf{r}_1, \uparrow), \dots, (\mathbf{r}_{N_\uparrow}, \uparrow), (\mathbf{r}_{N_\uparrow+1}, \downarrow), \dots, (\mathbf{r}_{N_\uparrow+N_\downarrow}, \downarrow)). \quad (5)$$

By renaming the integration variables,

$$\begin{aligned} \langle \hat{O} \rangle &= \frac{\sum_\sigma \int \Psi^*(\mathbf{R}) \hat{O}(\mathbf{R}) \Psi(\mathbf{R}) d\mathbf{R}}{\sum_\sigma \int \Psi^*(\mathbf{R}) \Psi(\mathbf{R}) d\mathbf{R}} \\ &= \frac{\int \Psi^*(\mathbf{R}) \hat{O}(\mathbf{R}) \Psi(\mathbf{R}) d\mathbf{R}}{\int \Psi^*(\mathbf{R}) \Psi(\mathbf{R}) d\mathbf{R}}. \end{aligned}$$

We further introduce the notation $\mathbf{R}^\uparrow \equiv (\mathbf{r}_1^\uparrow, \dots, \mathbf{r}_{N_\uparrow}^\uparrow) \equiv (\mathbf{r}_1, \dots, \mathbf{r}_{N_\uparrow})$ and $\mathbf{R}^\downarrow \equiv (\mathbf{r}_1^\downarrow, \dots, \mathbf{r}_{N_\downarrow}^\downarrow) \equiv (\mathbf{r}_{N_\uparrow+1}, \dots, \mathbf{r}_{N_\uparrow+N_\downarrow})$. Then the constraint that $\Psi(\mathbf{X})$ is antisymmetric with respect to the action of S_N is equivalent to the requirement that

$$\Psi(\mathbf{R}) := \Psi(\mathbf{R}^\uparrow, \mathbf{R}^\downarrow) \quad (6)$$

is an antisymmetric function with respect to \mathbf{R}^\uparrow and \mathbf{R}^\downarrow separately, but not necessarily antisymmetric across the spin-up and spin-down indices. We can thus work with $\Psi(\mathbf{R})$ instead of $\Psi(\mathbf{X})$ as long as we enforce this symmetry. If needed, the original antisymmetric function $\Psi(\mathbf{X})$ can be recovered by antisymmetrizing the following spin-dependent function:

$$\Psi(\mathbf{R}) \delta_{\sigma_1, \uparrow} \dots \delta_{\sigma_{N_\uparrow}, \uparrow} \delta_{\sigma_{N_\uparrow+1}, \downarrow} \dots \delta_{\sigma_{N_\downarrow}, \downarrow}. \quad (7)$$

2.2. Variational Monte Carlo

In the variational Monte Carlo (VMC) method, the goal is to find the variational minimizer Ψ of the energy functional

$$E[\Psi] = \frac{\int \Psi^*(\mathbf{R}) H \Psi(\mathbf{R}) d\mathbf{R}}{\int \Psi^*(\mathbf{R}) \Psi(\mathbf{R}) d\mathbf{R}}. \quad (8)$$

Given a wavefunction Ψ , which may not be normalized, we can estimate the high-dimensional integral in Eq. (8) via Markov-Chain Monte Carlo sampling. We first define the probability distribution associated with $\Psi(\mathbf{R})$ as $p(\mathbf{R}) = |\Psi(\mathbf{R})|^2 / \int |\Psi(\mathbf{R})|^2 d\mathbf{R}$. We then write the energy functional as a simple expectation over $p(\mathbf{R})$:

$$\frac{\int \Psi^*(\mathbf{R}) H \Psi(\mathbf{R}) d\mathbf{R}}{\int \Psi^*(\mathbf{R}) \Psi(\mathbf{R}) d\mathbf{R}} = \int_{\mathbb{R}^{Nd}} \frac{H \Psi(\mathbf{R})}{\Psi(\mathbf{R})} p(\mathbf{R}) d\mathbf{R} = \int_{\mathbb{R}^{Nd}} E_L(\mathbf{R}) p(\mathbf{R}) d\mathbf{R}, \quad (9)$$

where we have defined the so-called *local energy* associated with Ψ as

$$E_L(\mathbf{R}) = \frac{H \Psi(\mathbf{R})}{\Psi(\mathbf{R})}. \quad (10)$$

The general strategy for solving the molecular many-body problem (1) is then to parametrize a family of real wavefunctions $\Psi_\theta \in L^2$ on some parameter domain (usually \mathbb{R}^m , where m is the number of parameters) and combine the parameterization with Monte Carlo estimation to solve the resulting approximate eigenproblem. This is done by drawing a set of samples ξ_θ from $p_\theta(\mathbf{R})$ using Markov-Chain Monte Carlo and estimating the loss as

$$\mathcal{L}(\theta) = E[\Psi_\theta] \approx \frac{1}{|\xi_\theta|} \sum_{\mathbf{R} \in \xi_\theta} E_L(\mathbf{R}; \theta) = \tilde{\mathcal{L}}(\theta). \quad (11)$$

In statistical machine learning parlance, $\tilde{\mathcal{L}}$ is known as the empirical risk, and \mathcal{L} is known as the true or population risk.

3. Architectures

In this section, we describe the architectures for the neural network-based ansatzes that we explore. We first describe the FermiNet architecture, which consists of a (generalized) backflow layer that produces permutation equivariant features, followed by an antisymmetric layer to compute the wavefunction amplitude. We then describe the other architectures that we explore, which use different antisymmetric layers but all share the same generalized backflow layer used in the FermiNet.

The overall structure of the FermiNet may be described as a composition of a general $S_{N_\uparrow} \times S_{N_\downarrow}$ equivariant feature map (see the definition below) $\mathbf{Y}(\mathbf{R}) \equiv (\mathbf{Y}^\uparrow(\mathbf{R}), \mathbf{Y}^\downarrow(\mathbf{R}))$, with an antisymmetric layer constructed as a sum of products of determinants of orbital matrices $\Phi^{k\sigma}$:

$$\Psi(\mathbf{R}) = \sum_{k=1}^K \det \Phi^{k\uparrow}(\mathbf{Y}^\uparrow(\mathbf{R})) \det \Phi^{k\downarrow}(\mathbf{Y}^\downarrow(\mathbf{R})). \quad (12)$$

We describe the equivariant feature map and the orbital-determinant layer separately below.

3.1. Permutation equivariant features in FermiNet

The FermiNet feature map \mathbf{Y} is a map from particle positions $\mathbf{R} \in \mathbb{R}^{N \times d}$ to generalized coordinates $\mathbf{Y} \in \mathbb{R}^{N \times d'}$. While $d = 3$ is the dimension of the physical space, d' is the number of features and can be chosen arbitrarily. One particularly important symmetry of interest is given by the tensor product of the canonical permutation representations of S_{N_\uparrow} and S_{N_\downarrow} , defined by the action

$$(\pi \otimes \rho)(\mathbf{R}) = (\pi \otimes \rho)(\mathbf{R}^\uparrow, \mathbf{R}^\downarrow) = (\pi(\mathbf{R}^\uparrow), \rho(\mathbf{R}^\downarrow)) = (\mathbf{r}_{\pi(1)}^\uparrow, \dots, \mathbf{r}_{\pi(N_\uparrow)}^\uparrow, \mathbf{r}_{\rho(1)}^\downarrow, \dots, \mathbf{r}_{\rho(N_\downarrow)}^\downarrow). \quad (13)$$

The FermiNet map \mathbf{Y} is equivariant with respect to this action, i.e.

$$\mathbf{Y}((\pi \otimes \rho)(\mathbf{R})) = (\pi \otimes \rho)(\mathbf{Y}(\mathbf{R})). \quad (14)$$

To ensure that \mathbf{Y} is equivariant with respect to elements of $S_{N_\uparrow} \times S_{N_\downarrow}$ while incorporating information about the two-electron distances, the basic layer involves two streams: a “one-electron stream” that starts with the electron positions $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_N)$, and a “two-electron stream” that starts with the electron-electron displacements $\mathbf{r}_{ij} \equiv \mathbf{r}_i - \mathbf{r}_j$. The streams are averaged over the electrons, concatenated onto the one-electron stream, and a dense layer followed by a nonlinear activation function such as \tanh is applied. Residual connections [28] are also used between layers of the same shape for both streams.

More precisely, if $\mathbf{h}_i^{l\alpha}$ and $\mathbf{h}_{ij}^{l\alpha\beta}$ are the outputs of the one- and two-electron streams at layer l with spins $\alpha, \beta \in \{\uparrow, \downarrow\}$, then the concatenated vector for index i and spin α is

$$\mathbf{f}_i^{l\alpha} = \left(\mathbf{h}_i^{l\alpha}, \frac{1}{N_\uparrow} \sum_{j=1}^{N_\uparrow} \mathbf{h}_j^{l\uparrow}, \frac{1}{N_\downarrow} \sum_{j=1}^{N_\downarrow} \mathbf{h}_j^{l\downarrow}, \frac{1}{N_\uparrow} \sum_{j=1}^{N_\uparrow} \mathbf{h}_{ij}^{l\alpha\uparrow}, \frac{1}{N_\downarrow} \sum_{j=1}^{N_\downarrow} \mathbf{h}_{ij}^{l\alpha\downarrow} \right) \quad (15)$$

and the output of layer $l+1$ is given by the two streams

$$\begin{aligned} \mathbf{h}_i^{(l+1)\alpha} &= \tanh(V^l \mathbf{f}_i^{l\alpha} + \mathbf{b}^l) + \mathbf{h}_i^{l\alpha}, \\ \mathbf{h}_{ij}^{(l+1)\alpha\beta} &= \tanh(W^l \mathbf{h}_{ij}^{l\alpha\beta} + \mathbf{b}^l) + \mathbf{h}_{ij}^{l\alpha\beta}. \end{aligned} \quad (16)$$

As an initial pre-processing step, the electron positions \mathbf{R} are converted to “atomic coordinates” as

$$\mathbf{h}_i^{0\alpha} = \left(\mathbf{r}_i^\alpha - \mathbf{R}_1^{(a)}, \left| \mathbf{r}_i^\alpha - \mathbf{R}_1^{(a)} \right|, \dots, \mathbf{r}_i^\alpha - \mathbf{R}_M^{(a)}, \left| \mathbf{r}_i^\alpha - \mathbf{R}_M^{(a)} \right| \right), \quad (17)$$

$$\mathbf{h}_{ij}^{0\alpha\beta} = \left(\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta, \left| \mathbf{r}_i^\alpha - \mathbf{r}_j^\beta \right| \right), \quad (18)$$

which are invariant with respect to a simultaneous translation of the entire system. The explicit dependence on the absolute values enables the network to efficiently represent the derivative discontinuity due to the electron-nuclei cusp and electron-electron cusp, respectively [9,29].

Each map $(\mathbf{h}^{\uparrow}, \mathbf{h}^{\downarrow}) \mapsto (\mathbf{h}^{(l+1)\uparrow}, \mathbf{h}^{(l+1)\downarrow})$ is $S_{N_\uparrow} \times S_{N_\downarrow}$ equivariant due to the averaging procedure [25] in Eq. (15), and the map $\mathbf{R} \mapsto \mathbf{h}^0$ is parallel in the particle index i and thus equivariant as well. Therefore the map $\mathbf{R} \mapsto \mathbf{Y} \equiv (\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow) \equiv (\mathbf{h}^{L\uparrow}, \mathbf{h}^{L\downarrow})$, where L is the total number of one-electron layers, is also $S_{N_\uparrow} \times S_{N_\downarrow}$ equivariant. This is called a backflow map, which generalizes the original proposal of “backflow” by Feynman and Cohen [13].

Note that both \mathbf{Y}^\uparrow and \mathbf{Y}^\downarrow depend on the positions of all electrons. The index α in the notation \mathbf{Y}^α does not denote a dependence of \mathbf{Y} only on the corresponding α -spin inputs \mathbf{R}^α , but rather denotes the symmetry constraint of \mathbf{Y}^α as equivariant with respect to the action of S_{N_α} on \mathbf{R}^α and invariant with respect to the action of S_{N_β} on \mathbf{R}^β . Hence in this paper, we shall refer to the index $\{\uparrow, \downarrow\}$ in the expression $(\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow)$ as the *pseudospin* index. It has been shown constructively, though without explicit error bounds, that a simplified version of this construction [23] can approximate all and only the equivariant continuous functions.

3.2. Antisymmetric layer in FermiNet

Once the equivariant feature maps $\mathbf{Y} = (\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow)$ are generated, they are then used to generate K pairs of orbital matrices $(\Phi^{k\uparrow}(\mathbf{Y}^\uparrow), \Phi^{k\downarrow}(\mathbf{Y}^\downarrow))$. These orbital matrices are constructed using a set of per-pseudospin single particle orbitals $\varphi_i^{k\sigma}$ ($\sigma \in \{\uparrow, \downarrow\}$), which are defined by applying a simple dense layer to the equivariant features and multiplying by an exponential envelope function:

$$\varphi_i^{k\sigma}(\mathbf{y}_j^\sigma) = (\mathbf{w}_i^{k\sigma} \cdot \mathbf{y}_j^\sigma + \mathbf{b}_i^{k\sigma}) \sum_I d_{i,I}^{k\sigma} \exp\left(-\left|\Sigma_{i,I}^{k\sigma}(\mathbf{r}_j - \mathbf{R}_I^{(a)})\right|\right). \quad (19)$$

Here $\mathbf{w}_i^{k\sigma} \in \mathbb{R}^m$ and $b_i^{k\sigma} \in \mathbb{R}$ are the weights and biases of the dense layer, where m is the number of dimensions in the equivariant features \mathbf{y}_i^σ , which is chosen to be constant for all i, σ . The exponential envelopes are parameterized by $d_{i,I}^{k\sigma} \in \mathbb{R}$ and a matrix $\Sigma_{i,I}^{k\sigma} \in \mathbb{R}^{d \times d}$, which can also be set to a scaled identity matrix to simplify the ansatz. The parameters $\mathbf{w}, \mathbf{b}, \Sigma, \mathbf{d}$ are trainable and dependent on the pseudospin index σ . The exponential term ensures that the wavefunction is normalizable and that the support of each orbital does not extend too far away into the vacuum. While the orbital functions $\varphi_i^{k\sigma}$ are applied only to single feature vectors \mathbf{y}_j , the ansatz can express complex correlations because the feature vectors themselves depend on all of the particles in a complex way.

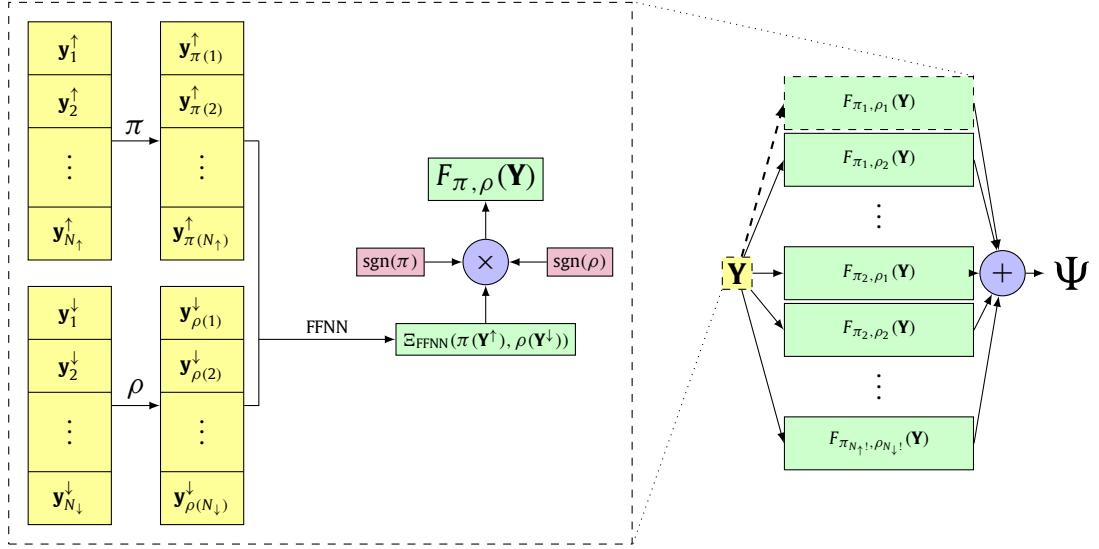


Fig. 1. The architecture for the generic antisymmetric layer. Left: calculation of the wavefunction contribution from a single pair of permutations π and ρ , denoted by $F_{\pi, \rho}(\mathbf{Y})$. Right: combination of contributions for all permutations π and ρ . This is implemented as a batch calculation over all pairs of permutations, but we show separate arrows for each pair of permutations to emphasize the factorial complexity of the operation.

The orbital matrices follow naturally from these single particle orbitals as

$$\Phi^{k\sigma}(\mathbf{Y}^\sigma) = \begin{pmatrix} \varphi_1^{k\sigma}(\mathbf{y}_1^\sigma) & \dots & \varphi_1^{k\sigma}(\mathbf{y}_{N_\sigma}^\sigma) \\ \vdots & \ddots & \vdots \\ \varphi_{N_\sigma}^{k\sigma}(\mathbf{y}_1^\sigma) & \dots & \varphi_{N_\sigma}^{k\sigma}(\mathbf{y}_{N_\sigma}^\sigma) \end{pmatrix}. \quad (20)$$

Once these orbital matrices are constructed, FermiNet creates an antisymmetric wavefunction using a sum of products of determinants:

$$\Psi(\mathbf{R}) = \sum_k \det \Phi^{k\uparrow} \det \Phi^{k\downarrow} \quad (21)$$

These determinants are similar to Slater determinants except that their orbitals are general equivariant functions of all of the input particles rather than simple single particle orbitals. Such determinants are thus referred to as generalized Slater determinants. In fact, the original FermiNet architecture [9] used a weighted sum of products of generalized Slater determinants by adding in a set of trainable parameters w_k and letting

$$\Psi(\mathbf{R}) = \sum_k w_k \det \Phi^{k\uparrow} \det \Phi^{k\downarrow}. \quad (22)$$

However, these trainable weights were removed by the original authors in their follow-up work [19], as they do not add extra expressiveness on top of the ability to tune the scale of the orbital matrices themselves. We follow the simplified construction.

3.3. Generic antisymmetric neural network layer

In order to assess the effectiveness of the antisymmetric layer of the FermiNet, we can replace the product of pseudospin determinant terms with a truly universal antisymmetric neural network layer. Let us first define the antisymmetrization operator on spin σ as

$$\mathcal{A}_\sigma[f](\mathbf{R}) \equiv \sum_{\pi_\sigma \in S_{N_\sigma}} (-1)^{\pi_\sigma} f(\pi_\sigma(\mathbf{R})), \quad (23)$$

where S_{N_σ} is the symmetric group on $\{1, \dots, N_\sigma\}$. Following the Leibniz formula for the determinant, the antisymmetric structure for the standard single-determinant FermiNet can be written as

$$\begin{aligned}
\Psi(\mathbf{R}) &= \det \Phi^\uparrow(\mathbf{Y}^\uparrow(\mathbf{R})) \cdot \det \Phi^\downarrow(\mathbf{Y}^\downarrow(\mathbf{R})) \\
&= \prod_{\sigma \in \{\uparrow, \downarrow\}} \left(\sum_{\pi_\sigma \in S_{N_\sigma}} (-1)^{\pi_\sigma} \varphi_1^\sigma(\mathbf{y}_{\pi_\sigma(1)}^\sigma) \cdots \varphi_{N_\sigma}^\sigma(\mathbf{y}_{\pi_\sigma(N_\sigma)}^\sigma) \right), \\
&\equiv \prod_{\sigma \in \{\uparrow, \downarrow\}} \mathcal{A}_\sigma [\Xi_{\text{FermiNet}}^\sigma](\mathbf{Y}^\sigma),
\end{aligned} \tag{24}$$

where we have defined the orbital product function $\Xi_{\text{FermiNet}}^\sigma(\mathbf{Y}^\sigma) = \prod_{i=1}^{N_\sigma} \varphi_i^\sigma(\mathbf{y}_i^\sigma, \theta_i^\sigma)$.

However, we can treat the interaction between pseudospins in a more universal way by considering just a single function $f(\mathbf{R}^\uparrow, \mathbf{R}^\downarrow)$ of all electron positions and antisymmetrizing f with respect to only the subsets of the input indices that correspond to the two spins:

$$\mathcal{A}_\uparrow \mathcal{A}_\downarrow [f](\mathbf{R}^\uparrow, \mathbf{R}^\downarrow) = \sum_{\pi_\uparrow \in S_{N_\uparrow}} \sum_{\pi_\downarrow \in S_{N_\downarrow}} (-1)^{\pi_\uparrow} (-1)^{\pi_\downarrow} f(\pi_\uparrow(\mathbf{R}^\uparrow), \pi_\downarrow(\mathbf{R}^\downarrow)). \tag{25}$$

The standard FermiNet can be written as the doubly antisymmetrized product of orbitals over all particles

$$\Psi_{\text{FermiNet}}(\mathbf{R}) = \mathcal{A}_\uparrow \mathcal{A}_\downarrow [\Xi_{\text{FermiNet}}](\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow), \tag{26}$$

where we define the product over all orbitals

$$\Xi_{\text{FermiNet}}(\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow) = \varphi_1^\uparrow(\mathbf{y}_1^\uparrow) \cdots \varphi_{N_\uparrow}^\uparrow(\mathbf{y}_{N_\uparrow}^\uparrow) \varphi_1^\downarrow(\mathbf{y}_2^\downarrow) \cdots \varphi_{N_\downarrow}^\downarrow(\mathbf{y}_{N_\downarrow}^\downarrow). \tag{27}$$

If we replace this all-particle product with a single feed-forward neural network (FFNN) denoted by $\Xi_{\text{FFNN}}(\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow)$, we arrive at an architecture that we refer to as the generic antisymmetric layer (GA):

$$\Psi_{\text{GA}}(\mathbf{R}) = \mathcal{A}_\uparrow \mathcal{A}_\downarrow [\Xi_{\text{FFNN}}](\mathbf{Y}^\uparrow, \mathbf{Y}^\downarrow). \tag{28}$$

This ansatz can be evaluated explicitly with $N_\uparrow! N_\downarrow!$ evaluations of Ξ_{FFNN} . Importantly, due to the equivariance property of \mathbf{Y} , we do not need to antisymmetrize the composed function $\Xi_{\text{FFNN}} \circ \mathbf{Y}$, but only the comparatively small Ξ_{FFNN} . We have found that we can achieve sufficient expressiveness with a very simple choice of Ξ_{FFNN} , further reducing the computational cost. In our experiments, we use a single hidden layer with $N_{\text{FFNN}} = 64$ nodes and a single application of the tanh activation function, followed by a linear combination operation. The cost of a single evaluation of Ξ_{FFNN} with a single hidden layer with N_{FFNN} nodes is $O(d(N_\uparrow + N_\downarrow)N_{\text{FFNN}})$, resulting in an overall cost of $O(d(N_\uparrow + N_\downarrow)N_\uparrow! N_\downarrow! N_{\text{FFNN}})$. We depict the construction of the GA layer in Fig. 1.

This construction is a universal replacement for the antisymmetry layer in the original FermiNet, due to the universality of neural networks to approximate functions of various desired smoothness classes with an appropriate choice of activation function [30–33]. In fact, the multiplication of two numbers can be approximated to arbitrary accuracy with just four neurons [34], and following equations (19) and (24), the FermiNet determinant is simply an antisymmetrized product of simple linear and exponential terms. In this work we choose Ξ_{FFNN} to be a one-hidden-layer feed-forward neural network for simplicity and efficiency, but one can choose any universal function class, e.g. residual neural networks.

3.4. Factorized antisymmetric neural network layer

Instead of replacing the entire antisymmetric layer of the FermiNet, we can consider replacing each product function $\Xi_{\text{FermiNet}}^\sigma$ in Eq. (24) with a feed-forward neural network, arriving at the ansatz

$$\Psi(\mathbf{R}) = \prod_{\sigma \in \{\uparrow, \downarrow\}} \mathcal{A}_\sigma [\Xi_{\text{FFNN}}^\sigma](\mathbf{Y}^\sigma). \tag{29}$$

We can relate this ansatz to the generic antisymmetric layer by assuming that Ξ_{FFNN} admits a functional low rank decomposition

$$\Xi_{\text{FFNN}}(\mathbf{Y}^\uparrow(\mathbf{R}), \mathbf{Y}^\downarrow(\mathbf{R})) \approx \sum_{k=1}^K \Xi_{\text{FFNN}}^{k\uparrow}(\mathbf{Y}^\uparrow(\mathbf{R})) \Xi_{\text{FFNN}}^{k\downarrow}(\mathbf{Y}^\downarrow(\mathbf{R})). \tag{30}$$

This is called the factorized antisymmetric layer of rank- K (FA- K). When $K = 1$, we recover the ansatz in Eq. (29). We treat this FA-1 layer, also depicted in Fig. 2, as an especially interesting case given the conjecture of [23] that a single generalized Slater determinant may be universal. We note that much like in the FermiNet pseudospin determinant terms, the electron positions of different spins do interact with each other in each FA pseudospin term due to the construction of the backflow (15).

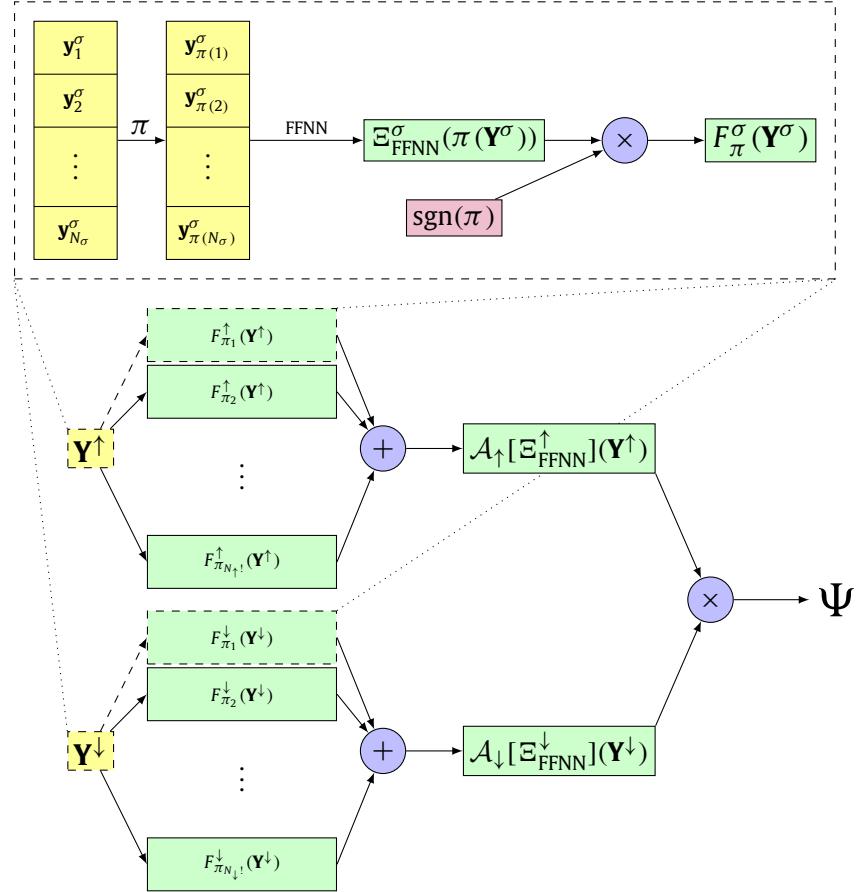


Fig. 2. The architecture for the factorized antisymmetric of rank 1 (FA-1). Top: calculation of the wavefunction contribution from a single pseudospin σ and permutation π , denoted by $F_\pi^\sigma(\mathbf{Y}^\sigma)$. Bottom: combination of contributions for all permutations π for both up and down pseudospin components. This is implemented as a batch calculation for each pseudospin, but we show separate arrows for each permutation to emphasize the factorial complexity of the operation. For FA- K this entire computation would be copied K times, with a different feed-forward-neural network in each copy, and the results would be added together.

We can evaluate the FA-1 layer using $N_\sigma!$ evaluations of Ξ_{FFNN}^σ for each pseudospin. The cost of a single evaluation of Ξ_{FFNN}^σ is $O(dN_\sigma N_{\text{FFNN}})$ operations for the matrix-vector multiplication, and the total cost of the explicit antisymmetrization for FA-1 is $O(d(N_\uparrow N_\uparrow! + N_\downarrow N_\downarrow!)N_{\text{FFNN}})$.

It is worth noting that both the factorized and the generic antisymmetric ansatzes have additional drawbacks beyond the obviously prohibitive factorial scaling. In our experiments with these ansatzes, we observed a great deal of numerical instability due to the massive numerical sign cancellation of the generally non-zero terms in the summations over the symmetric groups. We partially ameliorated this issue by performing the wavefunction evaluation in double precision instead of the more standard single precision (or even half precision) for modern deep learning, but even with this adjustment the numerical stability properties are far too unfavorable to scale these ansatzes as N_\uparrow and N_\downarrow grow large. Thus these ansatzes are certainly not intended to be used to directly approximate the ground state wavefunction of heavy atoms or large molecules, but are instead used in this paper as a diagnostic tool for better understanding the empirical performance of the FermiNet backflow and antisymmetry layers. More details about the practical effects of the numerical instabilities on our experiments are available in Appendix C.

3.5. Jastrow factors

Although the GA architecture can represent general antisymmetric functions on compact domains, we have found that without some mechanism of confining the support size of the wavefunction, the Monte Carlo sampling procedure often becomes unstable. In the standard FermiNet orbitals, this decay is handled by the simple exponential envelope terms in Eq. (19). In our generic antisymmetric layer, however, we have not directly included exponential decay terms in the antisymmetric part, so we require the presence of an additional decay term in the form of a Jastrow factor. We found that including an expressive Jastrow factor greatly increased the stability and accuracy of the ansatz, which suggests that the size of the wavefunction support and the behavior of the tails are of practical importance to the quality of the approximation.

In general, we may implement a Jastrow factor by multiplying an antisymmetric wavefunction ansatz by $\exp(J(\mathbf{R}^\uparrow, \mathbf{R}^\downarrow; \mathbf{R}^{(a)}))$, where the Jastrow factor J is a function of the electron positions \mathbf{R} and the nuclei locations $\mathbf{R}^{(a)}$. J must also be symmetric separately with respect to permutations of \mathbf{R}^\uparrow and \mathbf{R}^\downarrow in order to preserve the antisymmetry of the overall wavefunction. In order for $J(\mathbf{R})$ to capture the decay of the wavefunction, it will need to satisfy $J \rightarrow -\infty$ as $|\mathbf{R}| \rightarrow \infty$. The standard Jastrow form [16] which explicitly handles electron-nuclei, electron-electron, and electron-electron-nuclei terms is given by

$$J(\mathbf{R}; \mathbf{R}^{(a)}) = \sum_{\alpha \in \{\uparrow, \downarrow\}} \sum_{i=1}^{N_\alpha} \sum_{l=1}^M \chi_l \left(\left| \mathbf{R}_i^\alpha - \mathbf{R}_l^{(a)} \right| \right) + \sum_{\alpha \in \{\uparrow, \downarrow\}} \sum_{\beta \in \{\uparrow, \downarrow\}} \sum_{i=1}^{N_\alpha} \sum_{j=1}^{N_\beta} u^{\alpha\beta} \left(\left| \mathbf{R}_i^\alpha - \mathbf{R}_j^\beta \right| \right) \\ + \sum_{\alpha \in \{\uparrow, \downarrow\}} \sum_{\beta \in \{\uparrow, \downarrow\}} \sum_{i=1}^{N_\alpha} \sum_{j=1}^{N_\beta} \sum_l f_l^{\alpha\beta} \left(\left| \mathbf{R}_i^\alpha - \mathbf{R}_j^\beta \right|, \left| \mathbf{R}_i^\alpha - \mathbf{R}_l^{(a)} \right|, \left| \mathbf{R}_j^\beta - \mathbf{R}_l^{(a)} \right| \right), \quad (31)$$

where the functions $\{\chi_l\}$, $\{u^{\alpha\beta}\}$, and $\{f_l^{\alpha\beta}\}$ satisfy the desired behavior at infinity. One possibility is to use a simple one-body Jastrow from the first term above and let χ_l represent multiplication by a fixed constant for each nucleus, so that $\chi_l(r) = -a_l r$, with $a_l > 0$. Then we have

$$J(\mathbf{R}; \mathbf{R}^{(a)}) = \sum_{\alpha \in \{\uparrow, \downarrow\}} \sum_{i=1}^{N_\alpha} \sum_{l=1}^M \chi_l \left(\left| \mathbf{R}_i^\alpha - \mathbf{R}_l^{(a)} \right| \right) = \sum_{\alpha \in \{\uparrow, \downarrow\}} \sum_{i=1}^{N_\alpha} \sum_{l=1}^M -a_l \left| \mathbf{R}_i^\alpha - \mathbf{R}_l^{(a)} \right|. \quad (32)$$

Similarly, one could use the first two terms in Eq. (31) to form a simple two-body Jastrow, with a similar choice for the electron-electron interaction which is identical for the two spin species, i.e. $u^{\alpha\beta}(r) = -\gamma r$, with $\gamma > 0$. These approaches control the support size of the wavefunction, but do not allow much flexibility in the shape of the wavefunction tails outside of the asymptotic regime, where the wavefunction decay is known to be a simple isotropic exponential decay.

To build a more general Jastrow factor, we may leverage the generality of the FermiNet backflow construction to form the backflow-based Jastrow

$$J(\mathbf{R}; \mathbf{R}^{(a)}) = -\frac{1}{N} \sum_{\alpha \in \{\uparrow, \downarrow\}} \sum_{i=1}^{N_\alpha} \left| \mathbf{y}_{\text{Jastrow},i}^\alpha(\mathbf{R}) \right|. \quad (33)$$

For our numerical results involving the GA and FA- K architectures, we use this general Jastrow expression. The Jastrow factor needs to be able to grow small as the electron positions move far from the nuclei, which suggests the use of an unbounded activation function. To achieve this in practice, we simply swap out the tanh activation in Eq. (16) for an approximate GeLU activation [35],

$$\text{GeLU}(x) = 0.5x \left(1 + \tanh \left(\sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right) \quad (34)$$

which, like the hyperbolic tangent function, has the desirable property of being smooth everywhere.

3.6. Full determinant FermiNet

We also explore a variant of the FermiNet called the full determinant mode, which was mentioned in the original FermiNet paper [9] and implemented by the same authors in the JAX branch of the FermiNet repository [27]. Like the GA layer, the full determinant architecture does not assume a factorized form over the two pseudospins. In the full determinant mode, the single particle orbitals take exactly the same form as those used in the regular FermiNet architecture. The difference is that instead of using N_σ orbitals for each spin σ , we use N orbitals for both spins, where $N = N_\uparrow + N_\downarrow$. The up- and down-pseudospin orbital matrices are then concatenated into a square $N \times N$ matrix before taking the determinant. The new formula for the orbital matrices is

$$\Phi_{\text{full}}^{k\sigma}(\mathbf{Y}^\sigma) = \begin{pmatrix} \varphi_1^{k\sigma}(\mathbf{y}_1^\sigma) & \cdots & \varphi_1^{k\sigma}(\mathbf{y}_{N_\sigma}^\sigma) \\ \vdots & \ddots & \vdots \\ \varphi_N^{k\sigma}(\mathbf{y}_1^\sigma) & \cdots & \varphi_N^{k\sigma}(\mathbf{y}_{N_\sigma}^\sigma) \end{pmatrix}, \quad (35)$$

where the only difference from Eq. (20) is the change of the maximum orbital index from the pseudospin-specific N_σ to the total particle count N . Therefore $\Phi_{\text{full}}^{k\sigma}(\mathbf{Y}^\sigma)$ is a matrix of size $N \times N_\sigma$. The final ansatz is then generated as the sum of the determinants of the concatenated orbital matrices:

$$\Psi_{\text{FermiNet,full}}(\mathbf{R}) = \sum_{k=1}^K \det \left[\Phi_{\text{full}}^{k\uparrow}, \Phi_{\text{full}}^{k\downarrow} \right]. \quad (36)$$

The idea behind this construction is to provide a more flexible way to treat the interactions between the two pseudospin components. Importantly, because \mathbf{Y} is only equivariant with respect to permutations which exchange particles of the same spin, the concatenated determinant does not enforce an antisymmetry constraint between particles of opposite spins. We also note that it is possible to reconstruct the original FermiNet ansatz as a special case of the full determinant ansatz, by setting $\varphi_i^{k\sigma}(\mathbf{y}_j^\sigma) = 0$ whenever $\sigma = \uparrow$ and $i > N_\uparrow$ or $\sigma = \downarrow$ and $i \leq N_\uparrow$. In that case the full matrix becomes block-diagonal and the determinant factors into a simple product of pseudospin determinants [9]. We are particularly interested in the evaluating the performance of this full determinant mode when $K = 1$, which we refer to as the full single-determinant FermiNet.

4. Optimization

4.1. Gradient calculation

When computing parameter updates, estimating the gradient of $\mathcal{L}(\theta)$ by directly differentiating the empirical risk $\tilde{\mathcal{L}}(\theta)$ defined in Eq. (11) using an automatic differentiation framework is generally difficult due to the dependence of the Monte Carlo sampling on the parameters θ . However, the following standard unbiased estimate of the gradient of the true expected energy $E[\Psi_\theta] \equiv \mathcal{L}(\theta)$ is available for real wavefunctions [36]:

$$\begin{aligned} \partial_\theta \mathcal{L}(\theta) &= \frac{\int 2(\partial_\theta \log |\Psi_\theta|)(E_L(\mathbf{R}; \theta) - \mathcal{L}(\theta)) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \\ &\approx \frac{1}{|\xi_\theta|} \sum_{\mathbf{R} \in \xi_\theta} 2(\partial_\theta \log |\Psi_\theta|)(E_L(\mathbf{R}; \theta) - \tilde{\mathcal{L}}(\theta)) \end{aligned} \quad (37)$$

where ξ_θ are a set of samples from the density $p_\theta(\mathbf{R}) = |\Psi_\theta(\mathbf{R})|^2 / \int |\Psi_\theta|^2 d\mathbf{R}$. For completeness, the derivation of the gradient is provided in Appendix F.

The zero-variance principle [37] states that the eigenstates of the Hamiltonian (Eq. (1)) will have the same local energy everywhere. This improves the quality of the loss and gradient approximations as the training converges. In addition, this principle can make the variance of the local energy an attractive target for minimization. Indeed, variance minimization has an extensive history in the quantum Monte Carlo space [38–40]. Nonetheless, we follow the work of FermiNet [9,19] and only use energy minimization to optimize our wavefunctions.

4.2. Optimizer

The choice of efficient optimization algorithms for parameter updates in variational Monte Carlo has historically been a complex issue and is still under active debate (see e.g. [1,9,11,18,40–43]). Among these works, Ref. [9] provided evidence that the use of the Kronecker Factorized Approximate Curvature (KFAC) method [44] can be advantageous when compared to standard stochastic gradient descent-like methods used in the machine learning community such as Adam [45]. KFAC is a method for approximating natural gradient descent efficiently by preconditioning the gradient with an approximate inverse of the Fisher information matrix. Both the overall structure of KFAC and the extra steps required to apply KFAC to an unnormalized wavefunction are described succinctly in [9]. For the convenience of the reader we reproduce here an overview of these topics.

In the exact natural gradient descent method, the gradient of the loss function is multiplied by the inverse of the Fisher information matrix before using the gradient to make a parameter update [46]. This has the effect of taking the path of steepest descent not in Euclidean parameter space, but in the space of probability distributions defined by the model, with distance measured by the KL-divergence [47]. Concretely, updates in natural gradient descent take the form

$$\theta' = \theta - \eta \mathcal{F}^{-1} \nabla_\theta \mathcal{L}(\theta). \quad (38)$$

Here $\eta \in \mathbb{R}$ is the learning rate and \mathcal{F} is the Fisher information matrix defined as

$$\mathcal{F}_{ij} = \mathbb{E}_{p(\mathbf{R})} \left[\frac{\partial \log p(\mathbf{R})}{\partial \theta_i} \frac{\partial \log p(\mathbf{R})}{\partial \theta_j} \right]. \quad (39)$$

Note that in our case $p(\mathbf{R}) \propto |\Psi_\theta(\mathbf{R})|^2$, though the two are not equal as Ψ is not necessarily normalized. In fact, obtaining the Fisher information matrix requires a slightly different calculation in an unnormalized setting, which is usually referred to as stochastic reconfiguration [18]:

$$\mathcal{F}_{ij} \propto \mathbb{E}_{p(\mathbf{R})} \left[(\mathcal{O}_i - \mathbb{E}_{p(\mathbf{R})} [\mathcal{O}_i]) (\mathcal{O}_j - \mathbb{E}_{p(\mathbf{R})} [\mathcal{O}_j]) \right], \quad (40)$$

where $\mathcal{O}_i = \partial \log |\Psi_\theta(\mathbf{R})| / \partial \theta_i$. The equivalence of this formulation is proved in Appendix C of [9]. In the setting of quantum information geometry, the Fisher information matrix is proportional to, and perhaps more accurately viewed as, the Fubini-Study metric tensor or quantum geometric tensor [48].

Directly inverting the Fisher information matrix is infeasible for large models, as the matrix dimensions scale directly with the number of parameters. KFAC solves this problem by making two approximations to the Fisher matrix to allow its efficient inversion. The first is to assume that the Fisher entries for weights in different layers of the network are zero. This assumption reduces the Fisher matrix to a block diagonal form, so that inverting the remaining matrix only requires inverting each block independently. The second is based on the observation that the block corresponding to each layer of the network can be written as the mean-centered covariance of a Kronecker product of two vectors, one consisting of neuron activation values for the inputs to the layer and the other consisting of gradients of the loss with respect to the outputs of the layer. KFAC replaces this with the Kronecker product of the mean-centered covariance of the same vectors. As discussed in [44], this is a significant and theoretically unsupported approximation, but seems to work well in practice, at least in some use cases.

In our experiments, we rely on the JAX implementation of KFAC provided by the work of [9]. In using this implementation, we register all dense layers in our networks with KFAC, including those within the feed-forward neural networks of our generic antisymmetric and factorized antisymmetric layers. This ensures that we use the Kronecker product approximation of the Fisher matrix for all layers in the network, rather than defaulting to a simpler diagonal approximation.

5. Numerical experiments

In this section, we compare the previously described architectures on small atomic and molecular systems. All numerical experiments with the factorized and generic antisymmetric layers are performed using the VMCNet repository [49], which is based on the JAX framework [50]. In Appendix B, we demonstrate the comparability of the VMCNet repository with the JAX branch of the FermiNet repository [19]. Using JAX allows us to leverage the implementation of KFAC in [51], take advantage of the flexibility provided by JAX's clean functional style, and enjoy the performance benefits granted by its excellent out-of-the-box GPU utilization and just-in-time compilation. We used A100 GPUs on the Google Cloud Platform (GCP) for any calculations that required double precision, and GTX 2080TI GPUs with the Berkeley Research Computing (BRC) program for all other calculations.

To estimate energy values accurately after training, we ran pure MCMC for a large number of iterations without performing parameter updates, collecting samples every 10 iterations. We also estimated the integrated autocorrelation of the local energy during these evaluation runs in order to get a robust estimate of the standard error of our energy estimates. The hyperparameters we used, including the number of training and evaluation iterations, are listed in Appendix A. The gradient clipping and sampling procedures are described in Appendix G.

Throughout this section, we use the following standard notation to present our numerical results. All units are atomic units (a.u.) unless otherwise specified. The estimator of the energy used is the sample mean followed by, in parentheses, the standard error in the last digit(s) of the estimate. For example, $-54.58868(4)$ means a sample mean of -54.58868 a.u. with a standard error of approximately 4×10^{-5} a.u., and $-75.06314(13)$ means a sample mean of -75.06314 a.u. with a standard error of approximately 1.3×10^{-4} a.u.. The error of the energy is also measured by the percentage of the correlation energy recovered. The correlation energy is defined to be the difference between the Hartree-Fock energy and the exact ground state energy, so that recovering 0% of the correlation energy means that the calculation produces the Hartree-Fock energy, and recovering 100% means the calculation is exact. The correlation energy itself only contributes a tiny amount, usually less than 1%, to the ground state total energy, but capturing the correlation energy accurately is extremely important in chemistry.

5.1. Performance: atomic systems

We test our generic and factorized antisymmetric architectures on a few small atoms with nuclear charge from five to eight and compare these results to the results of FermiNet with 1 determinant, FermiNet with 1 full determinant, and FermiNet with 16 determinants. In Table 1, we compare the attained energies on these architectures after training with KFAC. These results are depicted as well in Fig. 3.

We find that the generic antisymmetric layer attains highly accurate energies when paired with the backflow-based Jastrow, achieving greater than 99.7% of the correlation energy. For the smallest systems, i.e. boron and carbon, the FermiNet-GA ansatz does at least as well as many-determinant FermiNet. For the larger systems, the performance of FermiNet-GA in our implementation began to suffer noticeably as we hit the limitations of our computational resources. For example, our result on oxygen for the generic antisymmetric architecture used only the simple one-body Jastrow in Eq. (32) and may not have reached the lowest energy that could be attained with additional training. We provide further discussion of the challenges with numerical stability and computational cost in Appendix C.

Interestingly, we do not see a gap in the attained energy between the factorized antisymmetric layer of rank 1 and single-determinant FermiNet for any system except for boron. To test the robustness of this result, we compare these two architectures on the Nitrogen atom with a variety of hyperparameter settings. We vary the depth of the equivariant feature map, the width of the one-electron and two-electron streams, and the depth and width of the feed-forward neural network

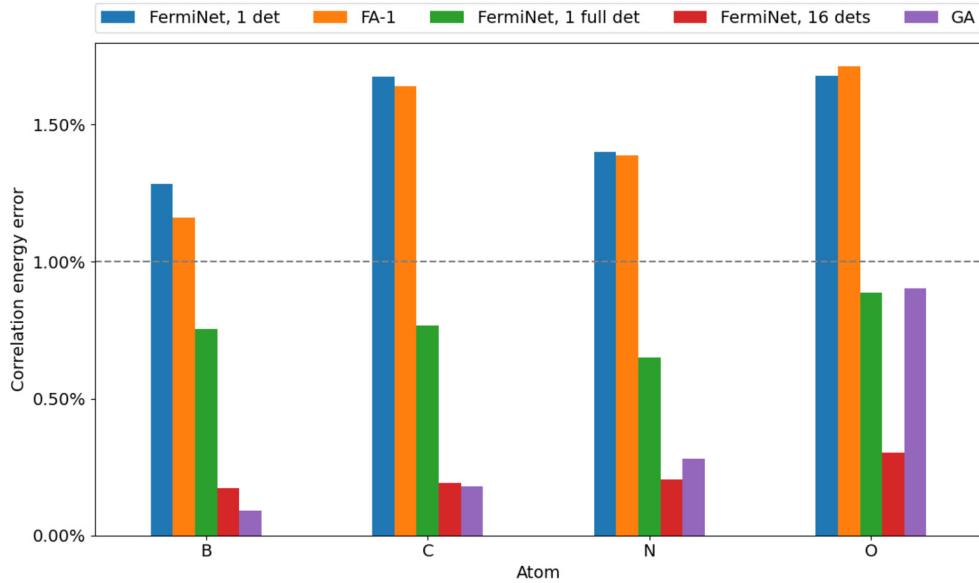


Fig. 3. Comparison of methods on atomic systems. FA stands for factorized antisymmetric layer, and GA stands for generic antisymmetric layer, as discussed in section 3. The 16 determinant FermiNet numbers are taken from Ref. [9]. The GA result on the oxygen atom uses the more restrictive one-body Jastrow, and the parameters may not be fully optimized due to the limitations of our resources. Dashed line indicates 1% of the error in correlation energy.

Table 1

Comparison of methods on atomic systems. FA stands for factorized antisymmetric layer, and GA stands for generic antisymmetric layer, as discussed in section 3. The 16 determinant FermiNet and Hartree-Fock results are taken from Ref. [9].

	FermiNet 1 det	FA-1	FermiNet 1 full det	FermiNet 16 dets [9]	GA	HF [9]	Reference [52]
B	-24.65236(3) 98.71(2)%	-24.65251(2) 98.84(2)%	-24.65300(3) 99.25(2)%	-24.65370(3) 99.83(3)%	-24.65380(2) 99.91(1)%	-24.53316	-24.65391
C	-37.84247(4) 98.33(3)%	-37.84252(3) 98.36(2)%	-37.84384(4) 99.23(3)%	-37.84471(5) 99.81(3)%	-37.84473(3) 99.82(2)%	-37.6938	-37.8450
N	-54.58662(5) 98.60(3)%	-54.58664(4) 98.61(2)%	-54.58800(8) 99.35(5)%	-54.58882(6) 99.79(3)%	-54.58868(4) 99.72(4)%	-54.4047	-54.5892
O	-75.06314(13) 98.32(5)%	-75.06305(6) 98.29(3)%	-75.06510(6) 99.11(2)%	-75.06655(7) 99.70(3)%	-75.06506(6) ^a 99.10(3)%	-74.8192	-75.0673

^a Due to the limitations of our computational resources, this result uses the simple but more restrictive one-body Jastrow from Eq. (32), and the parameters may not be fully optimized.

used in the factorized antisymmetric layer. The results are shown in Figs. 4 and 5. In all cases, the difference between the two methods is very small, and in particular is smaller than the difference between these methods and the GA method, when using default hyperparameter settings for GA.

To explore this trend further, we compare in Fig. 6 the factorized antisymmetric layers of rank 1 through 4 against the FermiNet with 1 through 4 determinants, again all on the nitrogen atom. We require two modifications to our optimization procedure to run these multi-determinant calculations. First, for all multi-determinant FermiNet calculations, we observe larger variance in the resulting energies between subsequent identical optimization runs. We describe this variance in more detail and with concrete data in Appendix D. To account for the variance, we report in Fig. 6 the best result of several runs. Second, when using 4 determinants, we find that the default VMCNet parameter initialization only captures 98.5–99% of the correlation energy, which is less than that captured by the 3-determinant FermiNet. We find that we can close this gap by adopting the default parameter initialization from the FermiNet code repository for the 4-determinant case. This entails using LeCun normal initializations rather than orthogonal initializations for the one-electron stream, the two-electron stream, and the dense layers in the orbital construction. The subtle dependence on the initialization and the run-to-run variance indicate that the VMC optimizer may get stuck at local minima, and that there is need for further improvement of the robustness of the optimization procedure.

We find that FA-K performs approximately equivalently to k -determinant FermiNet in all cases, and in most cases it performs slightly worse. This comparison is telling since replacing each generalized Slater determinant in the K -determinant FermiNet with an explicitly antisymmetrized feedforward neural network yields exactly the FA-K architecture. These results, together with the previously described comparisons between FA-1 and single-determinant FermiNet, strongly suggest that the reason that K -determinant FermiNet is not fully general is not due to the structure of the individual generalized Slater

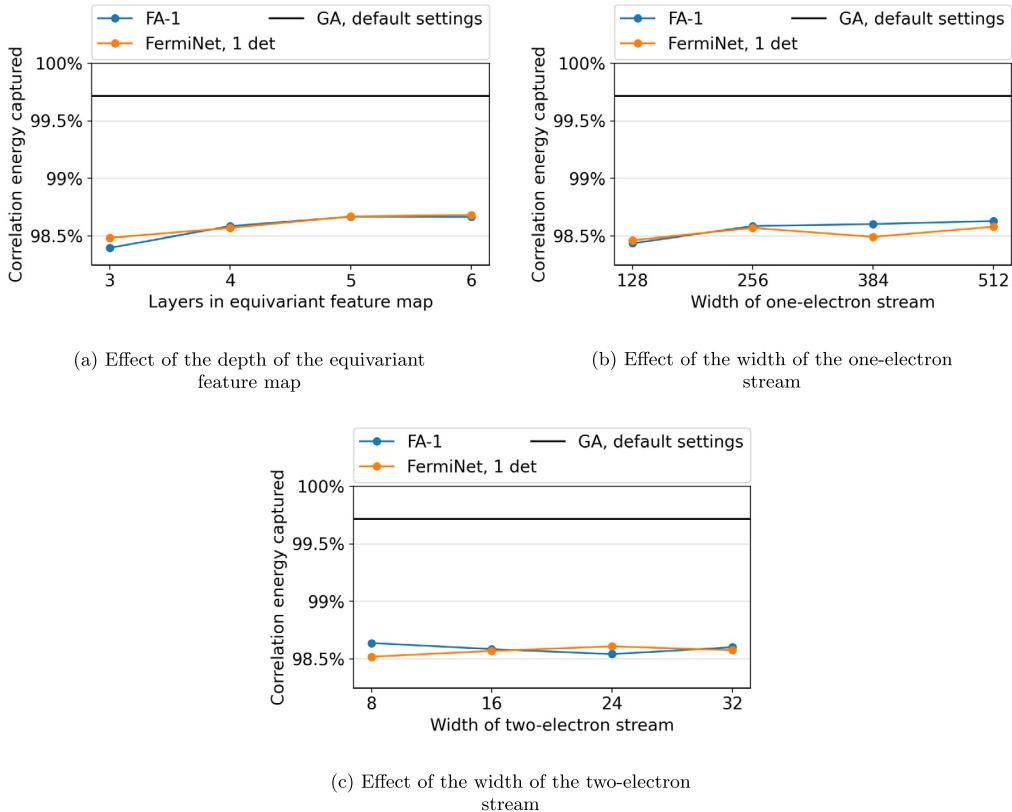


Fig. 4. Comparison of FA-1 and single-determinant FermiNet on the Nitrogen atom with varying hyperparameters for the equivariant feature map. The solid black line indicates the results of the GA architecture with default hyperparameter settings.

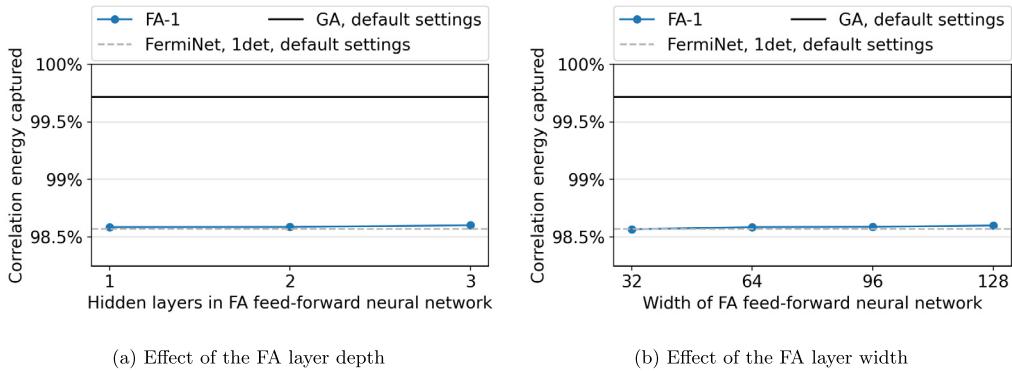


Fig. 5. Comparison of FA-1 and single-determinant FermiNet on the Nitrogen atom with varying hyperparameters for the feed-forward neural network used in the FA layer. The solid black line indicates the results of the GA architecture with default hyperparameter settings. The dotted line indicates the results of the single-determinant FermiNet architecture, also with default settings.

determinants, but rather due to the sum of products structure that is used to combine the generalized Slater determinants. This appears to be true even though the sum of products is taken with respect to the pseudospin components generated by the backflow rather than the original spins.

We also provide additional evidence in favor of the claim in [9] that KFAC provides an advantage over Adam when optimizing FermiNet-like architectures for small atoms and molecules. In Fig. 7 we provide a log-log plot of the correlation energy error during training of the generic and factorized antisymmetric architectures on the carbon atom. In this figure, the learning rate schedule for KFAC was chosen to be $5 \cdot 10^{-2} / (1 + 10^{-4}t)$. The learning rate schedule differed slightly for Adam, chosen instead to be $10^{-4} / (1 + 10^{-4}t)$. To determine the initial learning rate for both Adam and KFAC, we coarsely swept over a range of initial learning rates between $1e-4$ and $1e-1$ on the Carbon atom and picked the learning rate which resulted in the lowest final energies without encountering numerical instability or NaNs. The difference in the learning rates which we found were best for the two optimizers may be due to the different scale of the updates prior to the

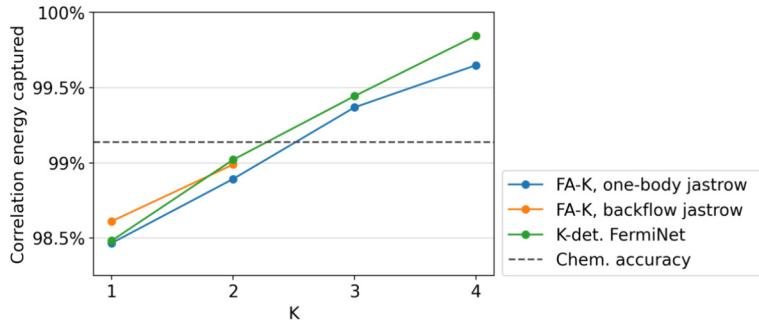


Fig. 6. Comparison of factorized antisymmetric layer of rank K with K -determinant FermiNet for $K = 1, 2, 3, 4$ on the nitrogen atom. Data for FA- K is presented with both the simple one-body Jastrow and the more expressive backflow based Jastrow, though results for the backflow Jastrow are limited to $K \leq 2$ due to numerical stability issues and computational resource constraints. Data for multi-determinant FermiNet represent the best of several runs to account for run-to-run variance.

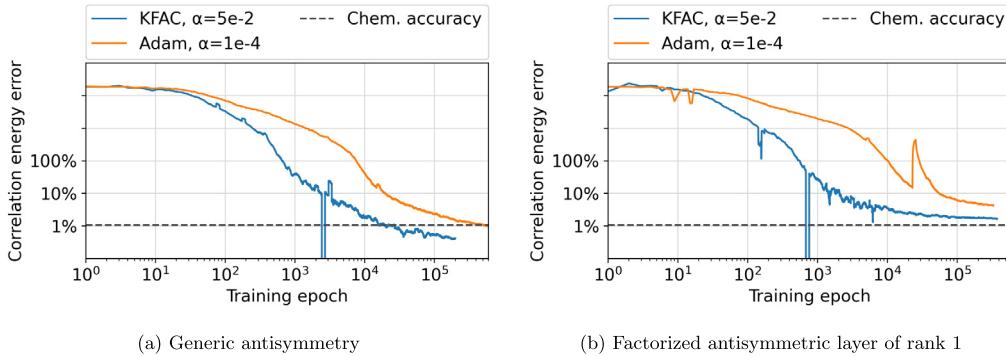


Fig. 7. Adam versus KFAC for the optimization of the generic and factorized antisymmetry architectures on the carbon atom. The proposed updates prior to the application of the learning rate may have different scales for the two optimizers, so we choose the largest stable initial learning rate for each from a coarse sweep of learning rates α with $\log_{10} \alpha \in [-4, -1]$. At each epoch, rolling averages of the previous 10% of training epochs are shown here for clarity. One epoch is one parameter update.

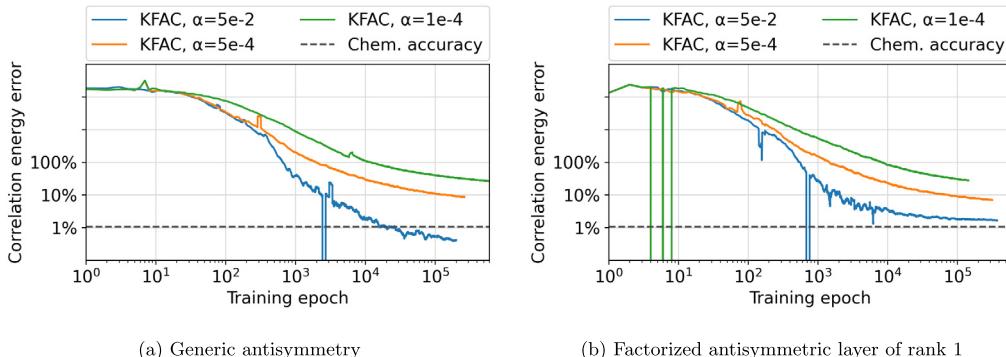


Fig. 8. KFAC at a few learning rates α for the optimization of the generic and factorized antisymmetry architectures on the carbon atom. At each epoch, rolling averages of the previous 10% of training epochs are shown here for clarity. One epoch is one parameter update. There is occasionally some initial instability within the first 10 or so epochs.

learning rate scaling. Fig. 8 shows that the choice of learning rate is quite important, and we generally found that in our experiments, using the highest consistently stable initial learning rate resulted in the lowest energies overall. When the learning rate is chosen in this way, KFAC reaches similar energy levels to Adam with as many as two orders of magnitude fewer epochs.

5.2. Performance: square H_4 model

The square H_4 model (Fig. 9) provides an interesting case study as a prototypical strongly correlated system [53]. Unlike the atomic case, the simple product of a pseudospin-up and pseudospin-down antisymmetry is inadequate to capture the ground state within a few percent of the correlation energy. In Fig. 10, we see that in both FA-1 and the standard single-

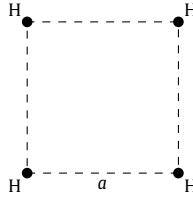


Fig. 9. Atomic configuration for the square H_4 model. The side length is a .

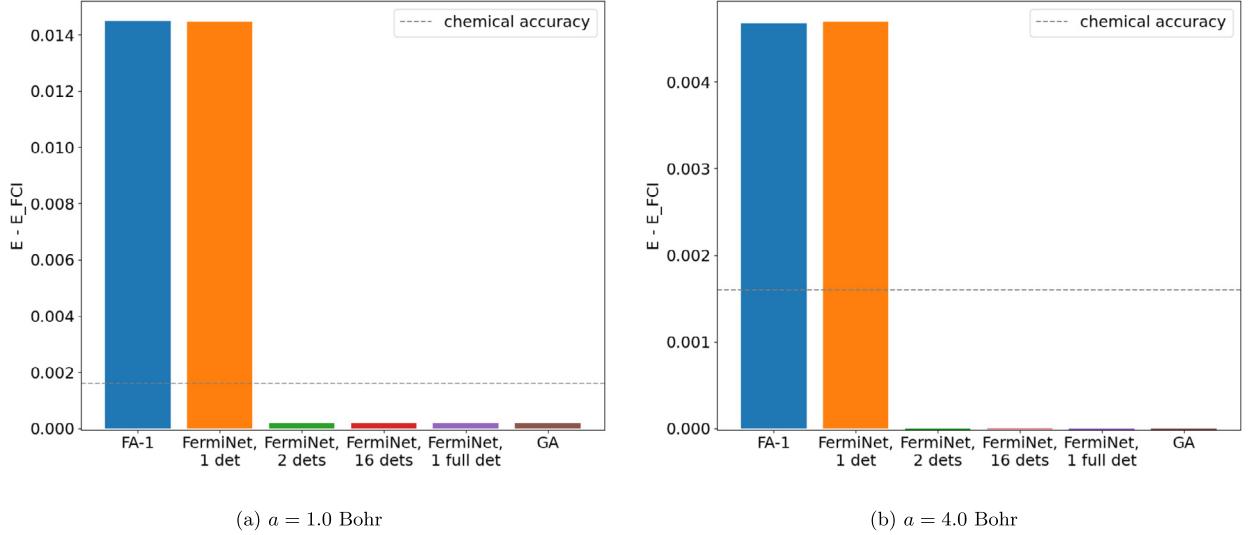


Fig. 10. Error of the total energy attained by various VMC ansatzes on the square H_4 model of different bond lengths. Details on the complete basis set extrapolation for FCI are included in Appendix E.

Table 2

Comparison of methods on the square H_4 model. Details on the complete basis set extrapolation for RHF/UHF/FCI are included in Appendix E.

a (Bohr)	RHF	UHF	FA-1	FermiNet 1 det	FCI
1.0	-1.2863	-1.3358	-1.424518(7)	-1.424531(6)	-1.4390
4.0	-1.7492	-2.0092	-2.029525(4)	-2.029502(4)	-2.0342
a (Bohr)	FermiNet 2 dets	FermiNet 16 dets	FermiNet 1 full det	GA	FCI
1.0	-1.438804(5)	-1.438796(5)	-1.438805(4)	-1.438799(4)	-1.4390
4.0	-2.034212(3)	-2.034208(3)	-2.034217(2)	-2.034218(3)	-2.0342

determinant FermiNet, the energy attained was significantly higher than that of any of the other neural network ansatzes tested here. We find excellent agreement between multiple-determinant FermiNet, full single-determinant FermiNet, and FermiNet-GA, agreeing within the estimated Monte-Carlo error. The failure of FA-1 to capture more of the ground-state energy than the standard single-determinant FermiNet again suggests that, at least for the small atomic and molecular systems modeled here, the FermiNet architecture is already very expressive for each pseudospin antisymmetry, even without an explicit Jastrow factor. The fact that even the addition of the general backflow-based Jastrow to the FA-1 architecture does not yield better results than FermiNet suggests that the lack of expressiveness of these simple “rank-one” product wavefunction ansatzes has to do with their nodal structure. See Table 2.

5.3. Comparison of nodal surfaces

Given a sufficiently general Jastrow correlation factor, the essential difficulty in the expressiveness of trial wavefunctions for quantum Monte Carlo methods lies in the accurate modeling of the nodal hypersurface [54]. We thus explore the nodal hypersurfaces generated by several of our ansatzes in Figs. 11 and 12, taking inspiration from [54]. In these figures, we fix the locations of all but one electron in the lithium and beryllium atoms and plot the nodal surface of the resulting one-body functions in the final electron position for four of our ansatzes: standard single-determinant FermiNet, FA-1, full

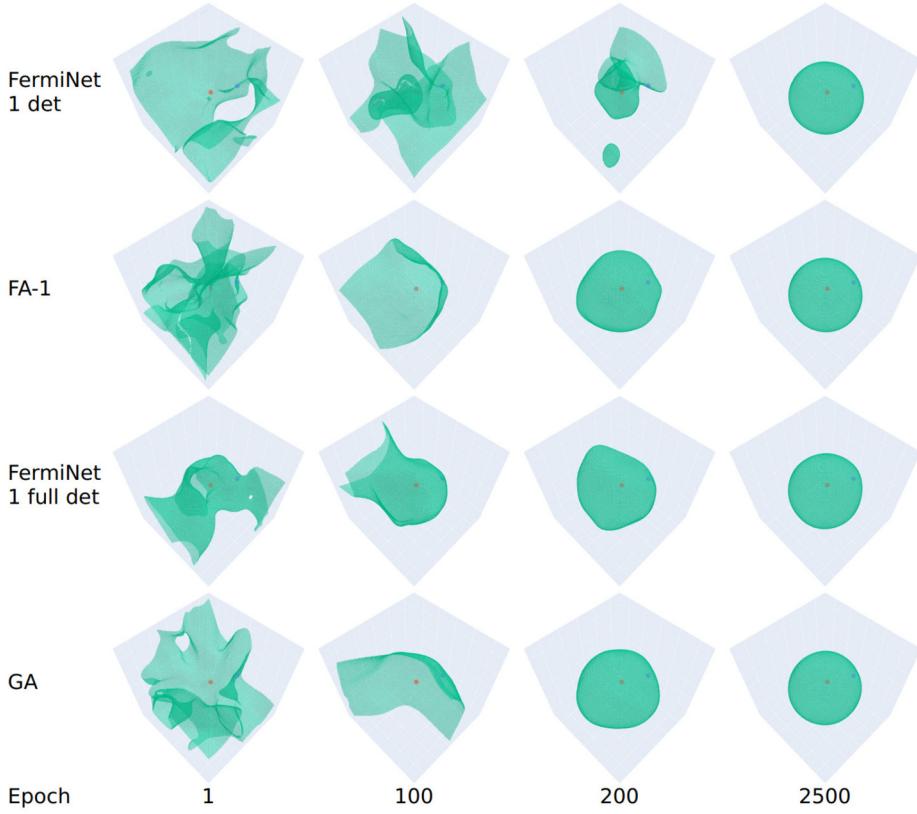


Fig. 11. Nodal surface cross-sections of various ansatzes on the lithium atom during training. The locations of two electrons are fixed, with one spin-up (same spin) electron in blue and one spin-down (opposite spin) electron in red. The fixed electrons are in random locations. The surfaces shown are produced by evaluating the sign of the wavefunction on the position of the remaining (spin-up) electron in the box $[-5, 5]^3$ on 150 points in each direction and using the `isosurface` graph object of the `Plotly` python graphing library. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

single-determinant FermiNet, and GA. This plotted nodal surface is thus a 3-dimensional cross-section of the full $(3N - 1)$ -dimensional nodal hypersurface of the many-body wavefunction, where for example for the beryllium atom $3N - 1 = 11$.

The nodal surface of the lithium wavefunction is essentially described by the two-particle antisymmetry between the two electrons of the same spin. In this two-particle regime, Ref. [23] shows the universality of the single generalized Slater determinant. Indeed, a generic antisymmetry of two particles can be exactly written as a single two-particle determinant with an appropriately general backflow, and so the architectures compared here are functionally equivalent in terms of their representation power. We see good agreement between the nodal surface cross-sections as early as epoch 2500 (Fig. 11).

However, in beryllium, we observe qualitative differences between the nodal surface cross-sections between the different architectures. If we choose random locations for the three fixed electron positions, we find that the nodal surface cross-sections look much like the smooth spheres in the lithium figure for all architectures. However, in Fig. 12 we choose the two opposite-spin electrons to be placed at $(0, \pm 2, 0)$, and we see that the nodal surface cross-sections for FermiNet and FA-1 (Fig. 12) appear to be the union of two smooth surfaces. We were able to confirm that these two surfaces originate from the product structure of the pseudospin terms by removing a pseudospin term and replotted the resulting nodal surface. On the other hand, the nodal surface cross-section obtained from GA and the full single-determinant FermiNet appear to consist of only one smooth surface. This difference aligns with our assertion that the product structure of FermiNet and FA-1 may limit their ability to represent the true nodal surface of the ground state wavefunction. A video is available with rotating views of the final cross-sections for all four wavefunction ansatzes [55].

Our study of the nodal surfaces in this section is importantly limited by the fact that we can only observe a 3-dimensional cross-section of the full nodal surface, so we are not able to directly draw conclusions about the global structure of the nodal surface when using only a single cross-section. Ideally, we could benchmark these plots against the ground truth of the nodal surface generated by the FCI wavefunction for this system. We found, however, that even the qualitative shape of the FCI nodal surface (not depicted here) depends strongly on the choice of the finite sized basis set, and is thus difficult to compare systematically to the VMC-derived wavefunctions.

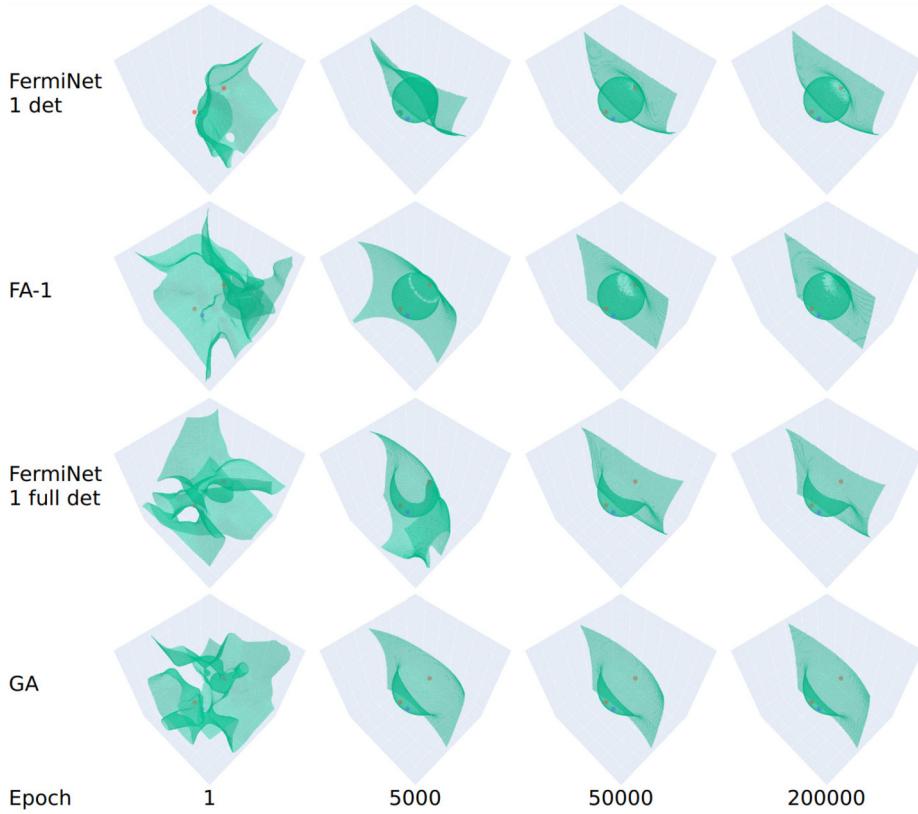


Fig. 12. Nodal surface cross-sections of various ansatzes on the beryllium atom during training. The locations of three electrons are fixed, with one spin-up (same spin) electron in blue and two spin-down (opposite spin) electron in red. The fixed spin-up electron is placed at $(2, 1, 0)$, and the two spin-down electrons are placed at $(0, \pm 2, 0)$. The surfaces shown are produced by evaluating the sign of the wavefunction on the position of the remaining (spin-up) electron in the box $[-5, 5]^3$ on 150 points in each direction and using the Isosurface graph object of the Plotly python graphing library. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

5.4. Nitrogen molecule: performance of the full determinant FermiNet

We finally provide a comparison of a few different FermiNet architectures on the nitrogen molecule at both the equilibrium bond length of 2.068 Bohr and a partially dissociated bond length of 4.0 Bohr. We are unable to test the FA and GA architectures on these systems within the constraints of our computational resources. However, we are able to test the standard FermiNet and the full determinant FermiNet. We find that the standard multi-determinant FermiNet consistently outperforms the standard single-determinant FermiNet. In addition, at equilibrium bond length, we find that the full single-determinant FermiNet outperforms the standard single-determinant FermiNet but does not outperform the standard 16-determinant (or greater) FermiNet. These results are both consistent with our findings on small atomic systems.

At a dissociating bond length of 4.0 Bohr, the N_2 molecule is a challenging strongly correlated system. Pfau et al. [9] demonstrate that the standard FermiNet architecture is not able to accurately model this system. We corroborate this finding on the VMCNet repository using 1 and 16 determinants, with results reported in Fig. 13. The energies may be found in Appendix I. In particular, Pfau et al. [9] report that for this system, the standard 64-determinant FermiNet does not outperform the 32-determinant FermiNet, which suggests that the energy does not approach the results obtained by the r_{12} -MR-ACPF method [57] as the number of standard determinants increases. However, we find that the full single-determinant FermiNet is able to outperform the standard 64-determinant FermiNet and come within chemical accuracy of the r_{12} -MR-ACPF value. This shows that some of the bottlenecks to the standard FermiNet's performance on particularly challenging systems is related to the interaction between pseudospin outputs of the backflow and determinant layers. We observed a non-trivial amount of run-to-run variance in our results, indicating that initialization and optimization methods for FermiNet-like architectures may require further investigation. However, we were able to replicate this result on several distinct optimization runs. This surprising result implies the need for further exploration into the benefits of the full determinant FermiNet for strongly correlated problems in quantum chemistry. Since the initial submission of this manuscript, a first step has been taken in this direction by Ren, Fu, and Chen [58], who show that the full 16-determinant FermiNet can outperform both the full single-determinant FermiNet and the r_{12} -MR-ACPF method.

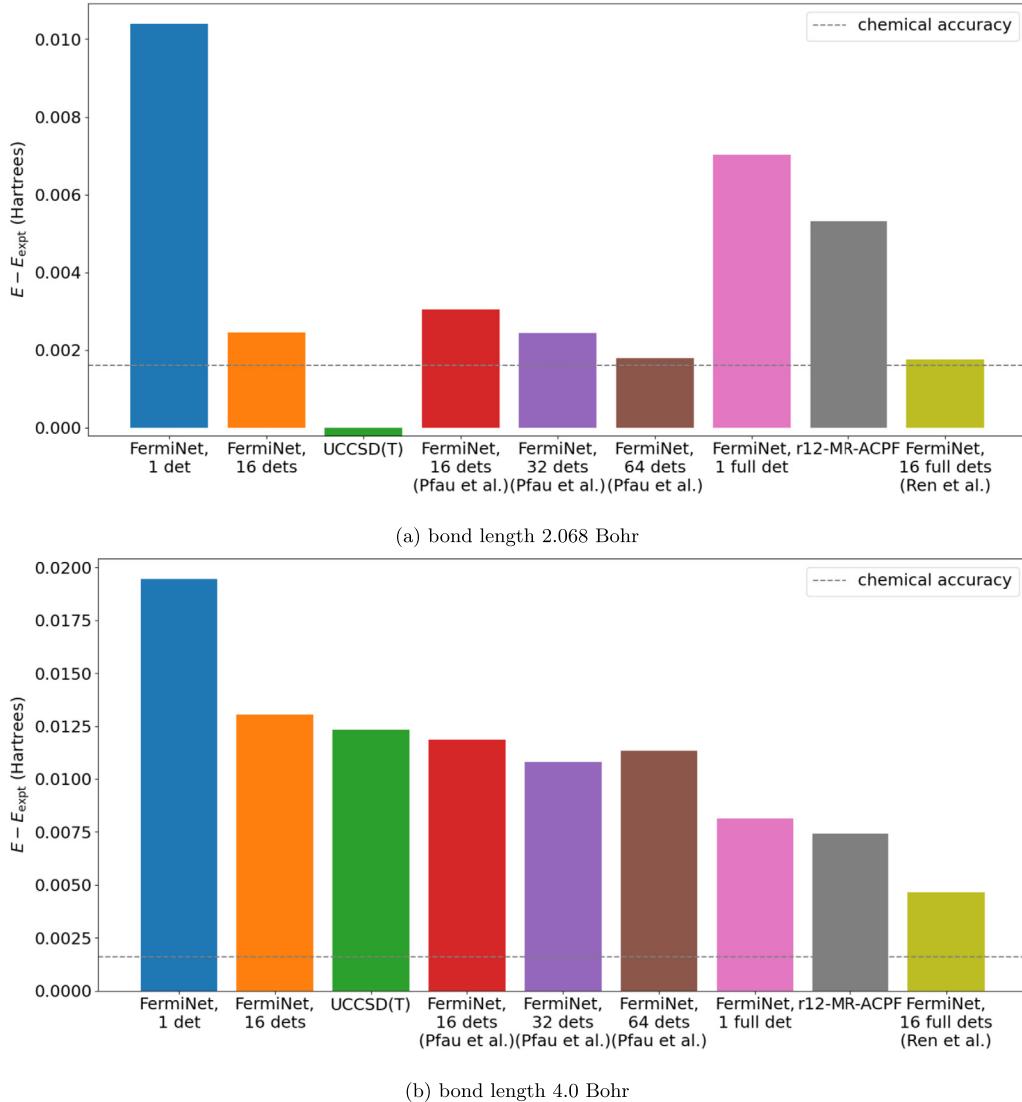


Fig. 13. Visual comparison of different FermiNet architectures on the nitrogen molecule at (a) bond length 2.068 Bohr (equilibrium) and (b) bond length stretched to 4.0 Bohr. Energies shown are the absolute energy difference (in Hartrees) between the attained energy (E) and the spectroscopic experimental fit (E_{expt}) recommended by Le Roy et al. [56]. The r_{12} -MR-ACPF computational reference is taken from [57]. The UCCSD(T) and the Pfau et al. energies were extracted from Figure 5 in reference [9]. The full 16-determinant FermiNet results were extracted from Figure 3a in [58]. The dashed line indicates chemical accuracy from the experimental fit. When a data point was not available at 2.068 Bohr, a linear interpolation between the two nearest data points was used.

6. Discussion

We show that explicitly antisymmetrized neural networks can be used to advance the understanding of the performance of neural network based VMC ansatzes. By replacing the antisymmetric layer in the FermiNet with a generic antisymmetrized neural network with relatively few nodes and a simple Jastrow factor, we find that the resulting FermiNet-GA structure is highly expressive and can yield accurate ground state energies (error of the correlation energy is less than 1%). On the other hand, if we replace each individual pseudospin determinant of FermiNet with an explicitly antisymmetrized neural network, the resulting FermiNet-FA- K structure does not outperform the K -determinant FermiNet. These observations suggest that the lack of expressiveness of the standard single-determinant FermiNet structure may be largely due to the product structure of the two pseudospin components of its determinant layer. This motivates us to investigate the “full determinant” mode of the FermiNet, which significantly improves the accuracy compared to both standard single-determinant FermiNet and FermiNet-FA-1.

One of our original motivations for developing FermiNet-GA was to resolve the challenges of the stretched nitrogen molecule as discussed in Ref. [9]. This is a challenging strongly correlated system which the standard FermiNet is unable to model accurately even with many determinants. We are currently unable to apply the GA layer to this system due to its

prohibitive computational cost. However, inspired by the success of the full determinant on smaller systems, we investigate its performance on the nitrogen molecule both at equilibrium and at a stretched bond length of 4.0 Bohr. At equilibrium, the full single-determinant FermiNet performs better than standard single-determinant FermiNet, but worse than standard multi-determinant FermiNet. However, in the stretched configuration, the full single-determinant FermiNet outperforms the standard 64-determinant FermiNet, achieving an energy within 0.4 kcal/mol of the r_{12} -MR-ACPF method [57]. This shows that it is essential to use the full determinant mode in order to get accurate energies from the FermiNet on this particular system. It is natural to ask whether there are other systems that exhibit a similar performance gap between the standard and the full determinant FermiNet, and whether it is possible to characterize the class of systems which exhibit such a gap. We hope that future work can clarify the answers to these questions.

In fact, since the initial submission of this manuscript, two preprints have been released which already shed further light on the utility of the full determinant construction. Casella et al. [59] use fermionic neural networks to study phase transitions of the homogeneous electron gas. They find that the use of full determinants (referred to as “dense determinants”) only marginally improves the performance relative to that of the standard FermiNet. This supports our finding that the magnitude of the advantage of full determinant FermiNet over the standard FermiNet can be highly system-dependent. Ren et al. [58] combine the FermiNet with diffusion Monte Carlo to obtain more accurate results on a variety of chemical systems. Their experiments are conducted using the full determinant mode, and the results on the nitrogen molecule corroborate our finding that the full determinant provides a significant advantage on this system. Their results show that the advantage is even greater when using 16 full determinants, as they obtain energies that are significantly lower than the energies from the r_{12} -MR-ACPF method.

As part of our work, we contribute a flexible, modular variational Monte Carlo repository called VMCNet [49], built on the JAX machine learning framework [50]. VMCNet is inspired by the JAX branch of the FermiNet repository [19] but uses the Flax API to facilitate developing new components and experimenting with different components of various architectures. We keep the model construction code in a separate submodule from the code for training, sampling, and evaluation, and we leverage simple asynchronous logging to enable monitoring of the training process. This work focuses on VMC simulation in the first quantization, but VMCNet can also be extended to simulate quantum systems in a second quantized form [60] as well.

We hope this work provides a first step towards understanding the expressiveness of these FermiNet-like architectures. Our results suggest the utility of diagnostic tools such as explicitly antisymmetrized neural networks for building such understanding. They also suggest the need to further explore the benefits of the full determinant mode of FermiNet. By further improvement upon the architecture and the optimization of the full determinant FermiNet, it may yet be possible to consistently achieve accurate results for a large class of physical and chemical systems of interest.

CRediT authorship contribution statement

Jeffmin Lin: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Gil Goldshlager:** Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Lin Lin:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by the Air Force Office of Scientific Research under award number FA9550-18-1-0095 (J.L.), by the NSF under Grant No. DMS-1652330 (G.G.), and by the Department of Energy under Grant No. DE-SC0017867, the CAMERA program and the SciDAC program (L.L.). L.L. is a Simons Investigator. We would like to acknowledge the use of computational resources at the Berkeley Research Computing (BRC) program at the University of California, Berkeley, the Google Cloud Platform (GCP), and the National Energy Research Scientific Computing Center (NERSC). We thank Giuseppe Carleo, Bryan Clark, Di Luo, James Stokes, Jiefu Zhang, Xiaojie Wu, and Fabian Faulstich for their helpful discussions. We also thank the hospitality of the American Institute of Mathematics (AIM) for the SQuaREs program “Deep learning and quantum Monte Carlo” in 2021.

Appendix A. Hyperparameters

In Table 3 we list the hyperparameters used in our runs for the KFAC optimizer. For the stretched N_2 geometry, to replicate the results reported by Ref. [9] as closely as possible, 4000 walkers were used instead of 2000, a two-electron stream width of 32 was used instead of 16, and when it resulted in lower energies, pretraining was also used for 1000 iterations.

Table 3

Table of hyperparameters for KFAC used during training.

Hyperparameter	Value
Dense nodes per layer in antisymmetrized part	64
Layers per ResNet in antisymmetrized part	2
One-electron stream width	256
Two-electron stream width	16
Number of layers in equivariant part	4
Kernel initializers for dense layers	orthogonal
Bias initializers for dense layers	random normal
Backflow activation function	tanh
ResNet antisymmetry activation function	tanh
Jastrow (backflow) activation function	gelu
Number of walkers	2000
Learning rate	$5 \cdot 10^{-2} / (1 + 10^{-4}t)$
Optimizer	KFAC
Threshold constant for local energy clipping	5.0
MCMC steps between updates	10
Training iterations (number of parameter updates)	$2 \cdot 10^5$
Evaluation iterations (samples collected every 10)	$2 \cdot 10^5$

Table 4

Comparison of the VMCNet repository with the FermiNet repository on the nitrogen atom, with 1, 2, and 4 determinants. Data for 2 and 4 determinants represent the best of several runs to account for run-to-run variance observed in both repositories.

Repository	1 det	corr %	2 det	corr %	4 det	corr %
VMCNet	-54.5864(1)	98.48(8)%	-54.58739(4)	99.02(2)%	-54.58891(4)	99.85(2)%
FermiNet	-54.58654(5)	98.56(3)%	-54.58711(6)	98.87(3)%	-54.58870(4)	99.73(4)%

Table 5Comparison of the VMCNet repository with the FermiNet repository on the H₄ square, with 1 and 2 determinants.

Repository	1 det	2 det
VMCNet	-1.424531(7)	-1.438804(5)
FermiNet	-1.424429(7)	-1.438796(5)

Appendix B. Code benchmarking

To demonstrate that our results for the original FermiNet are comparable to those reported by [9,19], we show that results obtained using the VMCNet repository are quantitatively comparable to that of the JAX branch of the FermiNet repository presented in [19] on several small systems. We compare the behavior on both the nitrogen atom and the square H₄ model (Fig. 9), using settings corresponding to the original FermiNet model in both repositories. For the nitrogen atom, we compare results with 1, 2, and 4 determinants, while for the H₄ square we compare results with just 1 and 2 determinants, since 2 determinants already capture essentially 100% of the correlation energy. All results presented here come from our own numerical experiments with either the VMCNet repository or the publicly available JAX branch of the FermiNet repository. Since VMCNet does not support Hartree-Fock based pretraining, we turned this feature off in the FermiNet repository to make the comparison fair. Turning off pretraining reduces the consistency of the FermiNet optimization on some systems. In particular, when using multiple determinants for the nitrogen atom, we found that some runs both of our own code and of the FermiNet code without pretraining get stuck in local minima and never reach the lowest energy possible. This phenomenon may merit further investigation. For now, to account for this run-to-run variance, we have taken the best of several runs for all multi-determinant experiments on the nitrogen atom.

Representative training graphs can be found for the nitrogen atom in Fig. 14 and for the H₄ square in Fig. 15. The values of the final energies obtained are presented in Tables 4 and 5, respectively. On both systems, the results of VMCNet are approximately equivalent to the results of FermiNet. The two repositories behave somewhat differently in the first 1,000 epochs of training, with VMCNet often optimizing more quickly in this regime. However, the optimization trajectories are largely indistinguishable by 10,000 epochs and the final energies achieved are within a small margin of error of each other in all cases.

Appendix C. Numerical stability and computational cost of the antisymmetric layer

One challenge we faced when training the generic antisymmetric architecture was the numerical sign cancellation near the nodal hypersurface. When computing FermiNet-GA in single precision, we invariably encountered NaNs (not-a-number).

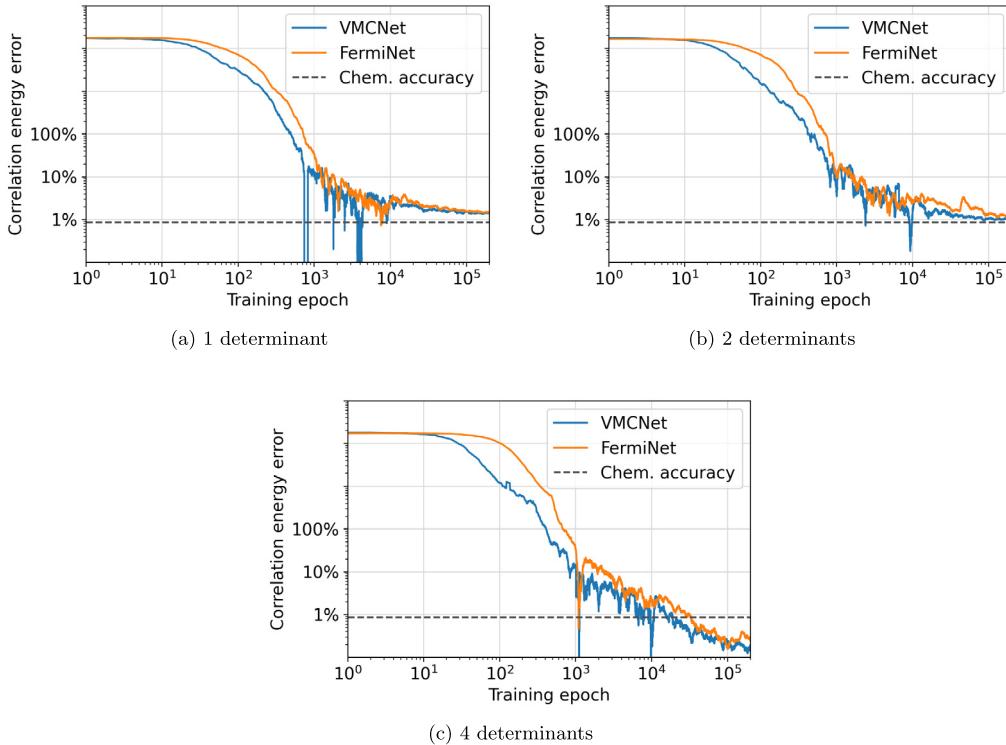


Fig. 14. Training runs on VMCNet and FermiNet repositories with the FermiNet architecture on the nitrogen atom. At each epoch, rolling averages of the previous 10% of training epochs are shown here for clarity. One epoch means one parameter update. Data for 2 and 4 determinants represent the best of several runs to account for run-to-run variance observed in both repositories.

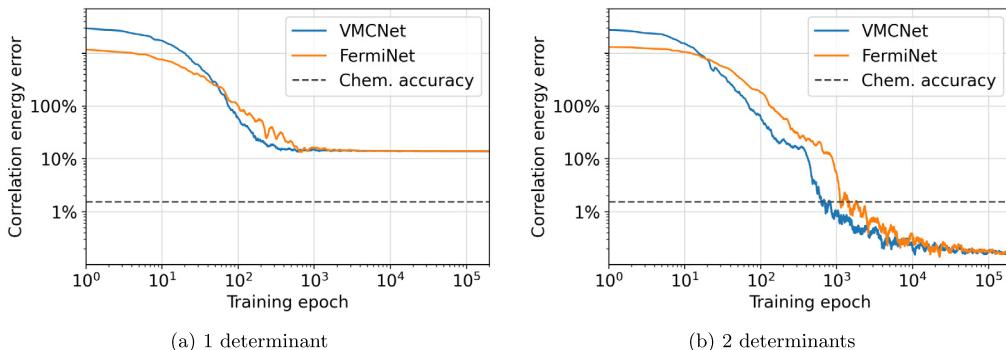


Fig. 15. Training runs on VMCNet and FermiNet repositories with the FermiNet architecture on the H4 square. At each epoch, rolling averages of the previous 10% of training epochs are shown here for clarity. One epoch means one parameter update.

Some investigation revealed that the computation of Ψ could yield slightly different results depending on whether it was calculated during a simple forward pass evaluation or a gradient calculation involving a forward and backward pass. Due to this numerical inconsistency, the Metropolis-Hastings procedure would sometimes sample points on or extremely close to the nodal hypersurface. To contend with this in our experiments, we used double precision end-to-end, i.e. converted all arrays to double precision. It is possible that a more efficient implementation might use double precision only in the antisymmetric layer or only when evaluating the local energy. The use of double precision led us to use A100 GPUs, which have significantly better performance for these higher precision calculations than consumer GTX GPUs. We used GTX 2080TI GPUs for our experiments which did not require double precision. Despite these powerful GPUs, the unfavorable scaling of the brute-force antisymmetry meant that we reached the limits of our group's resources with the calculations on the oxygen atom. On 4 A100 GPUs, training the FermiNet-GA architecture with the simplified Jastrow on the oxygen atom took 137 hours. In Fig. 16, we show the wall clock time used to train FA-1 and GA for boron through nitrogen, as well as the average time per iteration for standard single-determinant FermiNet, full single-determinant FermiNet, standard 16-determinant FermiNet, FA-1, and GA.

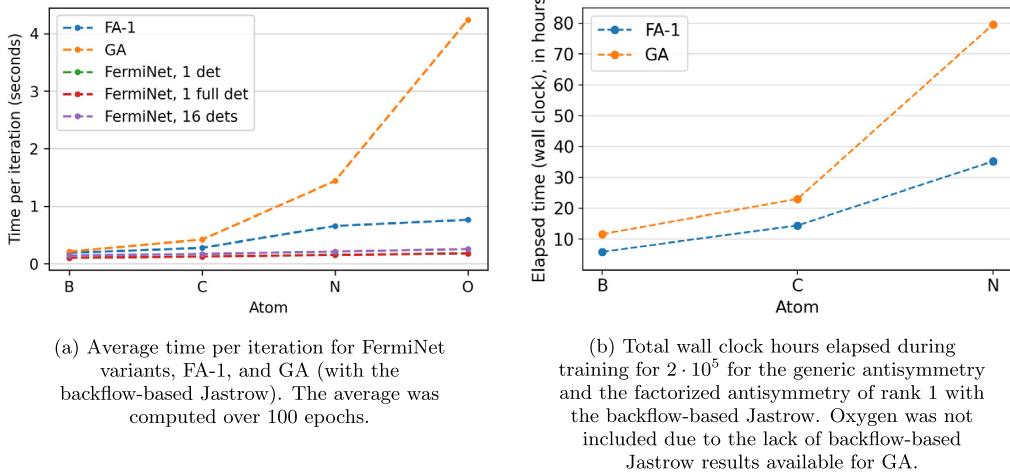


Fig. 16. Relative cost of various architectures on atomic systems during training. Tests were performed in double precision using 2 A100 GPUs.

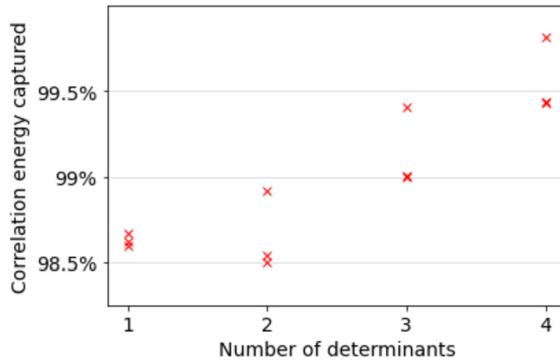


Fig. 17. Run-to-run variance of the k -determinant FermiNet on the Nitrogen atom, as a function of k . Results of three subsequent optimization runs are reported for each value of k .

Appendix D. Run-to-run variance

We report the run-to-run variance when optimizing the multi-determinant FermiNet structure on the Nitrogen atom. Subsequent optimization runs with all of the same hyperparameters often obtain energies with differences on the order of 0.5% of the correlation energy. This error is larger than the statistical error due to Monte Carlo sampling, which is on the order 0.05% of the correlation energy as reported in Table 1. To demonstrate this phenomenon more concretely, we report in Fig. 17 the results of three distinct optimization runs of the k -determinant FermiNet for each k from 1 to 4. The results for 4-determinant FermiNet use the initialization from the FermiNet repository as discussed in Section 5.1. Interestingly, the run-to-run variance is negligible for $k = 1$, but for each $k > 1$, we see that one of the three runs achieves a significantly better energy value than the other two. It is worth noting that these results do not incorporate the Hartree-Fock based pretraining used in [9].

Appendix E. Basis set extrapolation for the square H₄ model

The Hartree-Fock (HF) and full configuration interaction (FCI) values for the square H₄ model were extrapolated to the complete basis limit using cc-pvXz basis sets using PySCF [61,62]. For completeness, we reproduce the details of the extrapolation here.

The complete basis set Hartree-Fock energies were obtained by a fit to the function

$$E_{\text{HF}}(X) = E_{\text{HF}}(\text{CBS}) + a \exp(-bX), \quad (\text{E.1})$$

where $E_{\text{HF}}(X)$ is the Hartree-Fock energy computed with cc-pvXz and the parameters $E_{\text{HF}}(\text{CBS})$, a , and b are determined with a non-linear least-squares fit. Similarly, the complete basis set correlation energies are obtained by a fit to the function

$$E_{\text{corr}}(X) = E_{\text{corr}}(\text{CBS}) + aX^{-3}, \quad (\text{E.2})$$

where $E_{\text{corr}}(X)$ is the difference between the FCI and Hartree-Fock energies on the cc-pvXz basis and the parameters $E_{\text{corr}}(\text{CBS})$ and a are determined with a non-linear least-squares fit.

For RHF/UHF at bond length 1.0, we used $X = 2, 3, 4, 5$, as the orbital overlap matrix became too ill-conditioned for larger X . For RHF/UHF at bond length 4.0, we used $X = 5, 6, 8$. The FCI calculations were done using restricted Hartree-Fock (RHF) as the initial reference, and the correlation energy was computed as the difference between the FCI and RHF energies. For the extrapolation of the RHF-FCI correlation energy we used $X = 3, 4$. Due to the relative unreliability of the data points from the small double-zeta basis set and the cost of the quintuple-zeta basis set, these points were not included in the extrapolation for the correlation energy. Judging simply from the square root of the variance of the parameter fit, the basis set extrapolation error is at least two orders of magnitude larger than that of the Monte Carlo error in the estimates of the VMC-derived energies, so fewer significant digits are reported for the RHF/UHF/FCI results.

Appendix F. Gradient calculation

In this section we derive an unbiased estimate for the gradient of the expected energy of the wavefunction. Recall that the expected energy is given by

$$\mathcal{L}(\theta) = \frac{\int E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}}. \quad (\text{F.1})$$

For the purposes of the following derivation, we will let θ be a real number representing any parameter. The derivative of the integrand in the denominator with respect to θ is

$$\partial_\theta |\Psi_\theta|^2 = (\partial_\theta \Psi_\theta) \Psi_\theta^* + \Psi_\theta (\partial_\theta \Psi_\theta^*) = \left(\frac{\partial_\theta \Psi_\theta}{\Psi_\theta} + \frac{\partial_\theta \Psi_\theta^*}{\Psi_\theta^*} \right) |\Psi_\theta|^2, \quad (\text{F.2})$$

and the derivative of the integrand in the numerator with respect to θ is

$$\begin{aligned} \partial_\theta \left[E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 \right] &= \partial_\theta [\Psi_\theta^* H \Psi_\theta] \\ &= \partial_\theta \Psi_\theta^* H \Psi_\theta + \Psi_\theta^* H \partial_\theta \Psi_\theta \\ &= \frac{\partial_\theta \Psi_\theta^*}{\Psi_\theta^*} E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 + \Psi_\theta^* H \partial_\theta \Psi_\theta \end{aligned} \quad (\text{F.3})$$

To treat the latter term in the last expression above, we also take advantage of the following identity, which uses the essential self-adjointness of the Born-Oppenheimer Hamiltonian H [63]:

$$\int \Psi_\theta^* (H \partial_\theta \Psi_\theta) d\mathbf{R} = \int \partial_\theta \Psi_\theta (H \Psi_\theta^*) d\mathbf{R} = \int \frac{\partial_\theta \Psi_\theta}{\Psi_\theta} E_L(\mathbf{R}; \theta)^* |\Psi_\theta|^2 d\mathbf{R} \quad (\text{F.4})$$

Using these facts, the following calculation gives the derivative of the expected energy \mathcal{L} with respect to θ :

$$\begin{aligned} \partial_\theta \mathcal{L}(\theta) &= \partial_\theta \frac{\int E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \\ &= \frac{\partial_\theta \int E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} - \frac{\int E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \frac{\partial_\theta \int |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \\ &= \frac{\int ((\partial_\theta \Psi_\theta^* / \Psi_\theta^*) E_L(\mathbf{R}; \theta) + (\partial_\theta \Psi_\theta / \Psi_\theta) E_L(\mathbf{R}; \theta)^*) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \\ &\quad - \frac{\int E_L(\mathbf{R}; \theta) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \frac{\int (\partial_\theta \Psi_\theta^* / \Psi_\theta^* + \partial_\theta \Psi_\theta / \Psi_\theta) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}} \\ &= \frac{\int [(\partial_\theta \Psi_\theta^* / \Psi_\theta^*) (E_L(\mathbf{R}; \theta) - \mathcal{L}(\theta)) + (\partial_\theta \Psi_\theta / \Psi_\theta) (E_L(\mathbf{R}; \theta)^* - \mathcal{L}(\theta))] |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}}. \end{aligned} \quad (\text{F.5})$$

When the wavefunction is real, we may simplify this to

$$\partial_\theta \mathcal{L}(\theta) = \frac{\int 2(\partial_\theta \Psi_\theta / \Psi_\theta) (E_L(\mathbf{R}; \theta) - \mathcal{L}(\theta)) |\Psi_\theta|^2 d\mathbf{R}}{\int |\Psi_\theta|^2 d\mathbf{R}}. \quad (\text{F.6})$$

We may then use Monte Carlo sampling to estimate the gradient as

$$\partial_\theta \mathcal{L}(\theta) \approx \frac{1}{|\xi_\theta|} \sum_{\mathbf{R} \in \xi_\theta} 2(\partial_\theta \log |\Psi_\theta|) (E_L(\mathbf{R}; \theta) - \tilde{\mathcal{L}}(\theta)) \quad (\text{F.7})$$

where ξ_θ are a set of samples from the density $p_\theta(\mathbf{R}) = |\Psi_\theta(\mathbf{R})|^2 / \int |\Psi_\theta(\mathbf{R})|^2 d\mathbf{R}$.

Table 6

Numerical results for comparison of factorized antisymmetry of rank K with K -determinant FermiNet for $K = 1, 2, 3, 4$ on the nitrogen atom.

	FA- K , one-body Jastrow	corr%	FA- K , backflow Jastrow	corr%	K-det. FermiNet	corr%
$K=1$	54.58637(4)	98.47(2)	-54.58664(4)	98.61(2)	-54.5864(1)	98.48(8)
$K=2$	-54.58715(4)	98.89(2)	-54.58733(4)	98.99(2)	-54.58739(4)	99.02(2)
$K=3$	-54.58803(3)	99.37(1)	-	-	-54.58817(4)	99.44(2)
$K=4$	-54.58855(5)	99.65(3)	-	-	-54.58891(4)	99.85(2)

Table 7

Comparison of FermiNet architectures on the nitrogen molecule with at equilibrium bond length, 2.068 Bohr, and bond length stretched to 4.0 Bohr. When a data point was not available at 2.068 Bohr, a linear interpolation between the two nearest data points was used. The RHF result was extrapolated to the complete basis limit using cc-pvXz calculations in PySCF for $X = 2, 3, 4, 5$. The experimental result is computed from the MLR₄(6, 8) fitted potential curve recommended by the authors of [56].

	energy	
	2.068 Bohr	4.0 Bohr
RHF	-108.9940	-108.3101
FermiNet, 1 det	-109.5319	-109.1827
FermiNet, 16 det	-109.5398	-109.1891
FermiNet, 16 dets (Pfau et al. [9])	-109.5393	-109.1903
FermiNet, 32 dets (Pfau et al. [9])	-109.5399	-109.1913
FermiNet, 64 dets (Pfau et al. [9])	-109.5405	-109.1908
FermiNet, 1 full det	-109.5353	-109.1940
FermiNet, 16 full dets (Ren et al. [58])	-109.5405	-109.1975
UCCSD(T) (Pfau et al. [9])	-109.5425	-109.1898
r ₁₂ -MR-ACPF [57]	-109.5370	-109.1947
experiment [56]	-109.5423	-109.2021

Appendix G. Sampling and gradient clipping

We use the Metropolis-Hastings algorithm to sample electron configurations from the distribution defined by $\Psi(\mathbf{X})$. We use a gaussian proposal function with an isotropic step width, which we dynamically update throughout the optimization in order to keep the average acceptance ratio near a target value, for which we use 0.5. We maintain this ratio through the same simple scheme implemented in the JAX branch of the FermiNet repository [27], which increases the step width by a small amount if the acceptance ratio strays too far above the target, and similarly decreases it by a small amount if the ratio strays too far below the target. We perform such updates every 100 moves, averaging the acceptance ratio over the previous hundred steps in order to avoid overzealously updating the step width due to noise in the acceptance ratio.

In order to reduce the amount of correlation between the samples used for subsequent parameter updates, we take 10 walker steps between each gradient calculation and parameter update. While skipping steps theoretically does not produce a higher effective sample size than simply using every step, it is practically beneficial to skip steps because the local energy calculation required for a parameter update is significantly more computationally expensive than the wave function amplitude calculation required for each move. This means we can take a number of intermediate steps in order to produce significantly less correlated samples with a small computational overhead.

As is common in quantum Monte Carlo [64], in order to reduce the noise in the training process, we additionally clip the local energies calculated in each batch of samples to be closer to some estimator of the energy intended to reduce the effect of outliers in the gradient. Specifically, given a batch of local energies E_1, E_2, \dots, E_n , we calculate the median local energy E_M and then calculate the average deviation from the median (total variation) as

$$TV = \frac{1}{n} \sum_i |E_i - E_M|. \quad (G.1)$$

We then replace E_i with E_M whenever $|E_i - E_M| > 5 \cdot TV$. In practice, we have found that this produces a less noisy and more effective optimization process than including all of the unclipped local energies. During the final Monte Carlo evaluation of the energy after training, no local energy clipping is performed in order to avoid bias in the energy estimate.

Appendix H. Factorized antisymmetry versus FermiNet

We record the numerical results comparing FermiNet-FA- K to the standard K -determinant FermiNet in Table 6, also shown in Fig. 6. Due to the limits of our computational resources, we did not compute the $K = 3, 4$ results for FA- K with the backflow-based Jastrow.

Appendix I. Stretched nitrogen molecule

We record the numerical results comparing various FermiNet architectures on the nitrogen molecule with bond length 4.0 Bohr in Table 7. These results are also depicted in Fig. 13. For the nitrogen molecule, as we did not implement Hartree-Fock-based pretraining in VMCNet, we ported parameters from pretraining runs on the FermiNet repository, as we found that this increased the overall stability of the training procedure. As noted in Appendix A, we use 4000 walkers. The UCCSD(T) and Pfau et al. results were extracted from Figure 5 in [9].

References

- [1] G. Carleo, M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* 355 (2017) 602–606.
- [2] Y. Nomura, A.S. Darmawan, Y. Yamaji, M. Imada, Restricted Boltzmann machine learning for solving strongly correlated quantum systems, *Phys. Rev. B* 96 (2017) 205152.
- [3] K. Choo, G. Carleo, N. Regnault, T. Neupert, Symmetries and many-body excitations with neural-network quantum states, *Phys. Rev. Lett.* 121 (2018) 167204.
- [4] A. Nagy, V. Savona, Variational quantum Monte Carlo method with a neural-network ansatz for open quantum systems, *Phys. Rev. Lett.* 122 (2019) 250501.
- [5] D. Luo, B.K. Clark, Backflow transformations via neural networks for quantum many-body wave functions, *Phys. Rev. Lett.* 122 (2019) 226401.
- [6] J. Han, L. Zhang, W. E, Solving many-electron Schrödinger equation using deep neural networks, *J. Comput. Phys.* 399 (2019) 108929.
- [7] L. Yang, Z. Leng, G. Yu, A. Patel, W.-J. Hu, H. Pu, Deep learning-enhanced variational Monte Carlo method for quantum many-body physics, *Phys. Rev. Res.* 2 (2020), <https://doi.org/10.1103/physrevresearch.2.012039>.
- [8] J. Hermann, Z. Schätzle, F. Noé, Deep-neural-network solution of the electronic Schrödinger equation, *Nat. Chem.* 12 (2020) 891–897.
- [9] D. Pfau, J.S. Spencer, A.G.D.G. Matthews, W.M.C. Foulkes, Ab initio solution of the many-electron Schrödinger equation with deep neural networks, *Phys. Rev. Res.* 2 (2020), <https://doi.org/10.1103/physrevresearch.2.033429>.
- [10] K. Choo, A. Mezzacapo, G. Carleo, Fermionic neural-network states for ab-initio electronic structure, *Nat. Commun.* 11 (2020) 2368.
- [11] J. Stokes, J.R. Moreno, E.A. Pnevmatikakis, G. Carleo, Phases of two-dimensional spinless lattice fermions with first-quantized deep neural-network quantum states, *Phys. Rev. B* 102 (2020) 205122.
- [12] A. Szabo, N. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, McGraw-Hill, New York, 1989.
- [13] R.P. Feynman, M. Cohen, Energy spectrum of the excitations in liquid helium, *Phys. Rev.* 102 (1956) 1189.
- [14] L.F. Tocchio, F. Becca, A. Parola, S. Sorella, Role of backflow correlations for the nonmagnetic phase of the $t-t'$ Hubbard model, *Phys. Rev. B* 78 (2008) 041101.
- [15] W.M.C. Foulkes, L. Mitas, R.J. Needs, G. Rajagopal, Quantum Monte Carlo simulations of solids, *Rev. Mod. Phys.* 73 (2001) 33.
- [16] J. Gubernatis, N. Kawashima, P. Werner, *Quantum Monte Carlo Methods*, Cambridge Univ. Press, 2016.
- [17] J. Toulouse, R. Assaraf, C.J. Umrigar, Chapter fifteen - introduction to the variational and diffusion Monte Carlo methods, in: P.E. Hoggan, T. Ozdogan (Eds.), *Electron Correlation in Molecules – Ab Initio Beyond Gaussian Quantum Chemistry*, Advances in Quantum Chemistry, vol. 73, Academic Press, 2016, pp. 285–314.
- [18] F. Becca, S. Sorella, *Quantum Monte Carlo Approaches for Correlated Systems*, Cambridge University Press, 2017.
- [19] J.S. Spencer, D. Pfau, A. Botev, W.M.C. Foulkes, Better, faster fermionic neural networks, [arXiv:2011.07125 \[physics.comp-ph\]](https://arxiv.org/abs/2011.07125), 2020.
- [20] J. Han, Y. Li, L. Lin, J. Lu, J. Zhang, L. Zhang, Universal approximation of symmetric and anti-symmetric functions, [arXiv:1912.01765](https://arxiv.org/abs/1912.01765), 2019.
- [21] A. Sannai, Y. Takai, M. Cordonnier, Universal approximations of permutation invariant/equivariant functions by deep neural networks, [arXiv:1903.01939](https://arxiv.org/abs/1903.01939), 2019.
- [22] N. Keriven, G. Peyré, Universal invariant and equivariant graph neural networks, *Adv. Neural Inf. Process. Syst.* 32 (2019) 7092.
- [23] M. Hutter, On Representing (Anti)Symmetric Functions, Tech. Rep., DeepMind, London, UK, 2020, [arXiv:2007.15298](https://arxiv.org/abs/2007.15298).
- [24] M. Bachmayr, G. Dusson, C. Ortner, Polynomial approximation of symmetric functions, [arXiv:2109.14771](https://arxiv.org/abs/2109.14771), 2021.
- [25] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R.R. Salakhutdinov, A.J. Smola, Deep sets, in: *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3391–3401.
- [26] J. Kessler, F. Calcarechia, T.D. Kühne, Artificial neural networks as trial wave functions for quantum Monte Carlo, *Adv. Theory Simulations* 4 (2021) 2000269.
- [27] J.S. Spencer, D. Pfau, W.M.C. Foulkes, Ferminet: fermionic neural networks, <https://github.com/deepmind/ferminet>, 2020, commit, 27f90b.
- [28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [29] T. Kato, On the eigenfunctions of many-particle systems in quantum mechanics, *Commun. Pure Appl. Math.* 10 (1957) 151.
- [30] A.R. Barron, Approximation and estimation bounds for artificial neural networks, *Mach. Learn.* 14 (1994) 115.
- [31] A. Pinkus, Approximation theory of the mlp model in neural networks, *Acta Numer.* 8 (1999) 143–195.
- [32] D. Rolnick, M. Tegmark, The power of deeper networks for expressing natural functions, in: *International Conference on Learning Representations*, 2018.
- [33] D. Elbrächter, D. Perekrestenko, P. Grohs, H. Bölskei, Deep neural network approximation theory, [arXiv:1901.02220 \[cs.LG\]](https://arxiv.org/abs/1901.02220), 2021.
- [34] H.W. Lin, M. Tegmark, D. Rolnick, Why does deep and cheap learning work so well?, *J. Stat. Phys.* 168 (2017) 1223–1247.
- [35] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), [arXiv:1606.08415 \[cs.LG\]](https://arxiv.org/abs/1606.08415), 2016.
- [36] D. Bressanini, P.J. Reynolds, Between classical and quantum Monte Carlo methods: “variational” qmc, in: *Advances in Chemical Physics*, John Wiley & Sons, Ltd, 1999, pp. 37–64, <https://onlinelibrary.wiley.com/doi/10.1002/9780470141649.ch3>.
- [37] R.L. Coldwell, Zero Monte Carlo error or quantum mechanics is easier, *Int. J. Quant. Chem.* 12 (1977) 215, <https://onlinelibrary.wiley.com/doi/10.1002/qua.560120826>.
- [38] C.J. Umrigar, K.G. Wilson, J.W. Wilkins, Optimized trial wave functions for quantum Monte Carlo calculations, *Phys. Rev. Lett.* 60 (1988) 1719.

[39] P.R.C. Kent, R.J. Needs, G. Rajagopal, Monte Carlo energy and variance-minimization techniques for optimizing many-body wave functions, *Phys. Rev. B* 59 (1999) 12344.

[40] C.J. Umrigar, C. Filippi, Energy and variance optimization of many-body wave functions, *Phys. Rev. Lett.* 94 (2005) 150201.

[41] E. Neuscamman, C. Umrigar, G.K.-L. Chan, Optimizing large parameter sets in variational quantum Monte Carlo, *Phys. Rev. B* 85 (2012) 045103.

[42] L. Otis, E. Neuscamman, Complementary first and second derivative methods for ansatz optimization in variational Monte Carlo, *Phys. Chem. Chem. Phys.* 21 (2019) 14491.

[43] I. Sabzevari, A. Mahajan, S. Sharma, An accelerated linear method for optimizing non-linear wavefunctions in variational Monte Carlo, *J. Chem. Phys.* 152 (2020) 024111.

[44] J. Martens, R. Grosse, Optimizing neural networks with Kronecker-factored approximate curvature, in: Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 37, PMLR, Lille, France, 2015, pp. 2408–2417.

[45] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations, 2015.

[46] S.-i. Amari, Natural gradient works efficiently in learning, *Neural Comput.* 10 (1998) 251.

[47] S.-i. Amari, H. Nagaoka, *Methods of Information Geometry*, Oxford University Press, 2000.

[48] J. Stokes, J. Izaac, N. Killoran, G. Carleo, Quantum natural gradient, *Quantum* 4 (2020) 269.

[49] J. Lin, G. Goldshlager, L. Lin, VMCNet: Flexible, General-Purpose VMC Framework, Built on JAX, 2021.

[50] J. Bradbury, R. Frostig, P. Hawkins, M.J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: Composable Transformations of Python+NumPy Programs, 2018.

[51] https://github.com/deepmind/deepmind-research/tree/master/kfac_ferminet_alpha.

[52] S.J. Chakravorty, S.R. Gwaltney, E.R. Davidson, F.A. Parpia, C.F. p Fischer, Ground-state correlation energies for atomic ions with 3 to 18 electrons, *Phys. Rev. A* 47 (1993) 3649.

[53] K. Jankowski, J. Paldus, Applicability of coupled-pair theories to quasidegenerate electronic states: a model study, *Int. J. Quant. Chem.* 18 (1980) 1243.

[54] D.M. Ceperley, Fermion nodes, *J. Stat. Phys.* 63 (1991) 1237.

[55] <https://youtu.be/67SQXEUcYyY>.

[56] R.J. Le Roy, Y. Huang, C. Jary, An accurate analytic potential function for ground-state N2 from a direct-potential-fit analysis of spectroscopic data, *J. Chem. Phys.* 125 (2006) 164310.

[57] R.J. Gdanitz, Accurately solving the electronic Schrödinger equation of atoms and molecules using explicitly correlated (R12)-mr-ci: the ground state potential energy curve of N2, *Chem. Phys. Lett.* 283 (1998) 253.

[58] W. Ren, W. Fu, J. Chen, Towards the ground state of molecules via diffusion Monte Carlo on neural networks, 2022.

[59] G. Cassella, H. Sutterud, S. Azadi, N.D. Drummond, D. Pfau, J.S. Spencer, W.M.C. Foulkes, Discovering quantum phase transitions with fermionic neural networks, 2022.

[60] J.W. Negele, H. Orland, *Quantum Many-Particle Systems*, Westview, 1988.

[61] Q. Sun, T.C. Berkelbach, N.S. Blunt, G.H. Booth, S. Guo, Z. Li, J. Liu, J.D. McClain, E.R. Sayfutyarova, S. Sharma, S. Wouters, G.K.-L. Chan, Pyscf: the python-based simulations of chemistry framework, *WIREs Comput. Mol. Sci.* 8 (2018) e1340.

[62] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N.S. Blunt, N.A. Bogdanov, G.H. Booth, J. Chen, Z.-H. Cui, J.J. Eriksen, Y. Gao, S. Guo, J. Hermann, M.R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J.D. McClain, M. Motta, B. Mussard, H.Q. Pham, A. Pulkkin, W. Purwanto, P.J. Robinson, E. Ronca, E.R. Sayfutyarova, M. Scheurer, H.F. Schurkus, J.E.T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L.K. Wagner, X. Wang, A. White, J.D. Whitfield, M.J. Williamson, S. Wouters, J. Yang, J.M. Yu, T. Zhu, T.C. Berkelbach, S. Sharma, A.Y. Sokolov, G.K.-L. Chan, Recent developments in the pyscf program package, *J. Chem. Phys.* 153 (2020) 024109.

[63] M. Reed, B. Simon II, *Fourier Analysis, Self-Adjointness, Methods of Modern Mathematical Physics*, vol. 2, Academic Press, 1975.

[64] C.J. Umrigar, M.P. Nightingale, K.J. Runge, A diffusion Monte Carlo algorithm with very small time-step errors, *J. Chem. Phys.* 99 (1993) 2865.